

**PENINGKATAN PERFORMA METODE GAUSS NEWTON  
DENGAN CONJUGATE GRADIENT PADA MASALAH  
KLASIFIKASI BINER MENGGUNAKAN REGRESI  
LOGISTIK**

**SKRIPSI**

**OLEH:  
SILVI PUSPITA SARI  
NIM. 210601110060**



**PROGRAM STUDI MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG  
2025**

**PENINGKATAN PERFORMA METODE GAUSS NEWTON  
DENGAN CONJUGATE GRADIENT PADA MASALAH  
KLASIFIKASI BINER MENGGUNAKAN REGRESI  
LOGISTIK**

**SKRIPSI**

**Diajukan Kepada  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
untuk Memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh  
Silvi Puspita Sari  
NIM. 210601110060**

**PROGRAM STUDI MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG  
2025**

**PENINGKATAN PERFORMA METODE GAUSS NEWTON  
DENGAN CONJUGATE GRADIENT PADA MASALAH  
KLASIFIKASI BINER MENGGUNAKAN REGRESI  
LOGISTIK**

**SKRIPSI**

**Oleh:  
Silvi Puspita Sari  
NIM. 210601110060**

Telah Disetujui Untuk Diuji

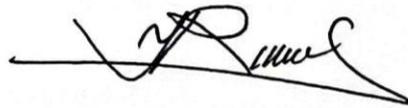
Malang, 27 Mei 2025

Dosen Pembimbing I



Dr. Mohammad Jamhuri, M.Si.  
NIP. 19810502 200501 1 004

Dosen Pembimbing II



Erna Herawati, M.Pd.  
NIPPPK. 19760723 2023211 2 006

Mengetahui,  
Ketua Program Studi Matematika



Dr. Elly Susanti, M.Sc.  
NIP. 19741129 200012 2 005

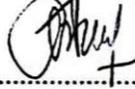
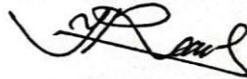
**PENINGKATAN PERFORMA METODE GAUSS NEWTON  
DENGAN CONJUGATE GRADIENT PADA MASALAH  
KLASIFIKASI BINER MENGGUNAKAN REGRESI  
LOGISTIK**

**SKRIPSI**

**Oleh:  
Silvi Puspita Sari  
NIM. 210601110060**

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)  
Tanggal 18 Juni 2025

Ketua Penguji : Dr. Usman Pagalay, M.Si  
Anggota Penguji 1 : Ari Kusumastuti, M.Pd., M.Si  
Anggota Penguji 2 : Dr. Mohammad Jamhuri, M.Si  
Anggota Penguji 3 : Erna Herawati, M.Pd

  
.....  
  
.....  
  
.....  
  
.....

Mengetahui,  
Ketua Program Studi Matematika  
  
  
Dr. Elly Susanti, M.Sc  
NIP. 19741129 200012 2 005

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini

Nama : Silvi Puspita Sari  
NIM : 210601110060  
Program Studi : Matematika  
Fakultas : Sains dan Teknologi  
Judul Skripsi : Peningkatan Performa Metode Gauss-Newton dengan  
conjugate gradient pada masalah klasifikasi biner  
menggunakan regresi logistik

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri, bukan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai pemikiran saya, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan di halaman terakhir. Apabila dikemudian hari terbukti skripsi ini adalah hasil jiplakan atau tiruan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 18 Juni 2025



Silvi Puspita Sari  
NIM. 210601110060

## **MOTO**

"Sesungguhnya Allah tidak akan membebani seorang hamba melampaui batas kemampuannya."

## **PERSEMBAHAN**

Dengan penuh rasa syukur ke hadirat Allah Swt., saya persembahkan skripsi ini kepada kedua orang tua tercinta, Bapak Masrukin dan Ibu Lasmi, yang telah memberikan cinta, doa, serta dukungan tanpa henti sepanjang perjalanan hidup saya. Terima kasih atas segala pengorbanan, ketulusan, dan kasih sayang yang tak pernah putus hingga saya dapat menyelesaikan studi ini. Saya memohon kepada Allah Swt. agar senantiasa melimpahkan kesehatan, kebahagiaan, dan keberkahan usia kepada Bapak dan Ibu. Semoga karya sederhana ini menjadi wujud bakti dan kebanggaan untuk kalian, serta menjadi amal kebaikan di hadapan Allah Swt

## KATA PENGANTAR

Segala puji dan syukur atas kehadiran Allah SWT dengan segala rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan dengan baik. Shalawat dan salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW, yang telah membimbing umat manusia keluar dari zaman jahiliah menuju cahaya terang yaitu agama islam. Skripsi ini disusun sebagai bagian dari pemenuhan sks untuk menyelesaikan matakuliah studi Matematika pada Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Selesainya penyusunan skripsi ini tidak lepas dari dorongan bermacam pihak, sebab itu diucapkan terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, S.Pd., M.Si., selaku ketua Progam Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Mohammad Jamhuri, M.Si, selaku pembimbing I yang selalu membimbing penulis dengan segala pengalaman serta memberikan segala arahan.
5. Erna Herawati, M.Pd., selaku pembimbing II yang telah sabar memberikan bimbingan, motivasi, saran, dan pengetahuan selama menyusun proposal skripsi.
6. Dr. Usman Pagalay, M.Si., selaku ketua penguji dalam ujian skripsi yang telah memberikan saran dan arahan yang bermanfaat bagi penulis.
7. Ari Kusumastuti M.Si., M.Pd., selaku anggota penguji 1 yang telah memberikan saran dan arahan yang bermanfaat bagi penulis.
8. Muhammad Khudzaifah, M.Si., selaku dosen wali yang telah sabar memberikan bimbingan, motivasi, saran, serta pengetahuan selama menyusun proposal skripsi.
9. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi Universitas Islam Maulana Malik Ibrahim Malang.
10. Kedua Orang tua serta keluarga yang telah memberikan dukungan dan motivasi serta do'a.

11. Seluruh mahasiswa angkatan 2021 dan sahabat-sahabat penulis yang selalu mendukung dan mendoakan antar sesama.
12. Ucapan terima kasih juga saya sampaikan kepada seseorang yang kehadirannya sangat berarti dalam hidup saya, Inggit Aditya Pratama. Terima kasih atas segala kontribusi, baik tenaga, waktu, maupun materi, yang telah diberikan selama proses penulisan karya tulis ini. Kehadiranmu menjadi tempat berlabuh, teman berbagi dalam suka dan duka, pendamping yang senantiasa memberi semangat di saat lelah, dan penghibur di kala sedih. Terima kasih telah menjadi bagian dari perjalanan ini, mendengarkan keluhan, serta memberikan dorongan untuk terus berjuang dan tidak menyerah. Semoga Allah senantiasa memberikan keberkahan dalam setiap langkah, doa, dan perjalanan hidup yang kita lalui ke depannya

Semoga Allah SWT membalas segala kebaikan mereka dengan balasan yang setimpal. Dengan adanya skripsi ini, semoga dapat memberikan manfaat kepada pembaca dan berkontribusi pada perkembangan ilmu pengetahuan.

Malang, 18 Juni 2025

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PENGAJUAN</b> .....	<b>ii</b>
<b>HALAMAN PERSETUJUAN</b> .....	<b>iii</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN</b> .....	<b>v</b>
<b>MOTO</b> .....	<b>vi</b>
<b>PERSEMBAHAN</b> .....	<b>vii</b>
<b>KATA PENGANTAR</b> .....	<b>viii</b>
<b>DAFTAR ISI</b> .....	<b>x</b>
<b>DAFTAR TABEL</b> .....	<b>xii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiii</b>
<b>DAFTAR SIMBOL</b> .....	<b>xiv</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xv</b>
<b>ABSTRAK</b> .....	<b>xvi</b>
<b>ABSTRACT</b> .....	<b>xvii</b>
<b>مستخلص البحث</b> .....	<b>xviii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah.....	6
1.6 Definisi Istilah.....	7
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>9</b>
2.1 Klasifikasi Biner.....	9
2.2 Regresi Logistik.....	11
2.3 Metode Gauss-Newton.....	17
2.4 Metode Conjugate Gradien.....	20
2.5 Integrasi Al-Qur'an dalam Prespektif Islam.....	26
2.6 Kajian Topik Dengan Teori Pendukung.....	30
<b>BAB III METODE PENELITIAN</b> .....	<b>32</b>

3.1	Jenis Penelitian .....	32
3.2	Data dan Sumber Data.....	32
3.3	Struktur Data.....	32
3.4	Tahapan Penelitian .....	33
3.4.1	Persiapan Data .....	33
3.4.2	Pelatihan Model Regresi Logistik.....	34
3.4.3	Implementasi Metode Gauss-Newton .....	35
3.4.4	Implementasi Metode Conjugate Gradient .....	36
3.4.5	Evaluasi Model.....	37
3.4.6	Flowchart.....	39
3.4.7	Perbandingan dengan Penelitian Sebelumnya .....	41
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>		<b>42</b>
4.1	Hasil.....	42
4.2	Inisialisasi Bobot.....	46
4.3	Hitung Fungsi Loss .....	49
4.4	Gauss Newton .....	50
4.5	Bangun Matriks Jacobian dan Matriks RHS D.....	52
4.6	Penyelesaian Sistem dengan Conjugate Gradient.....	54
4.7	Analisis Kurva Pembelajaran .....	62
4.8	Hasil Evaluasi Model pada Data Uji.....	64
4.9	Kurva Perbandingan dengan Metode Optimasi Dasar .....	67
4.10	Perbandingan dengan Metode Dasar.....	68
4.11	Perbandingan Hasil Penelitian dengan Penelitian Sebelumnya .....	69
4.12	Kajian Islam .....	70
<b>BAB V PENUTUP.....</b>		<b>75</b>
5.1	Kesimpulan.....	75
5.2	Saran.....	76
<b>DAFTAR PUSTAKA.....</b>		<b>78</b>
<b>LAMPIRAN .....</b>		<b>81</b>
<b>RIWAYAT HIDUP.....</b>		<b>88</b>

## DAFTAR TABEL

Tabel 4.1	Cuplikan 6 Data yang Diambil dengan 2 Fitur dengan Korelasi Tinggi .....	43
Tabel 4.2	Tabel Normalisasi 6 Data dengan 2 Fitur .....	46
Tabel 4.3	Perbandingan Nilai <i>Training Loss</i> pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih <i>Loss</i> Antara GN, Adam, dan GD.....	62
Tabel 4.4	Perbandingan Nilai <i>Training Accuracy</i> pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih <i>Accuracy</i> Antara GN, Adam, dan GD.....	62
Tabel 4.5	Perbandingan Nilai <i>Validation Loss</i> pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih <i>Loss</i> Antara GN, Adam, dan GD.....	63
Tabel 4.6	Perbandingan Nilai <i>Validation Accuracy</i> pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih <i>Loss</i> Antara GN, Adam, dan GD.....	63
Tabel 4.8	Tabel Confusion Matriks.....	64
Tabel 4.9	Perbandingan Performa dengan Menggunakan Metode GN, ADAM, dan GD pada Dataset Ionsphere.....	68
Tabel 4.10	Perbandingan dengan Penelitian Sebelumnya .....	69

## DAFTAR GAMBAR

Gambar 3.1 Flowchart Penelitian .....	40
Gambar 4.1 Presentasi <i>Good</i> dan <i>Bad</i> pada Data Ionosphere.....	42
Gambar 4.2 Perbandingan Nilai <i>Loss</i> pada Data Training dan Validasi Selama 100 Epoch .....	64
Gambar 4.3 Perbandingan Nilai <i>Loss</i> pada Data Training dan Validasi Selama 100 epoch.....	64
Gambar 4.4 Perbandingan Loss Data Training dan Validasi antara Metode GN, GD, dan ADAM.....	67
Gambar 4.5 Perbandingan kurva akurasi pada data Training dan Validasi Antara Metode Dasar GN, ADAM, GD.....	67

## DAFTAR SIMBOL

$\mathbb{R}^n$	: Ruang fitur berdimensi $n$
$X$	: Matriks fitur
$Y$	: Matriks label
$y_i$	: Label kelas
$\vec{x}_i$	: Vektor yang berisi nilai dari fitur-fitur variabel yang tersedia
$m$	: Jumlah sampel (data) dalam dataset
$n$	: Jumlah fitur (variabel) untuk setiap sampel
$\vec{\theta}$	: Koefisien bobot
$\theta_0$	: Bobot bias yang bekerja bersama nilai konstan 1
$h_i$	: Fungsi sigmoid
$z_i$	: Kombinasi linier
$\epsilon$	: Error satu data
$L$	: Fungsi biaya
$\nabla L(\vec{\theta})$	: Gradien dari fungsi biaya
$\eta$	: <i>Learning rate</i>
$\Delta \vec{\theta}$	: Vektor perubahan parameter
$A^T$	: Transpos dari matriks $A$
$\beta$	: Koefisien skalar
$\alpha$	: Koefisien langkah
$\vec{r}$	: Residual
$\vec{v}$	: <i>Euclidean inner product</i>
$\vec{w}$	: <i>Euclidean inner product</i>
$\vec{d}$	: Arah pencarian

## DAFTAR LAMPIRAN

Lampiran 1 Data Ionosphere .....	81
Lampiran 2 Hasil Normalisasi Seluruh Data.....	82
Lampiran 3 <i>Script Phyton</i> Metode Gauss Newton .....	80

## ABSTRAK

Sari, Silvi Puspita. 2025. **Peningkatan Performa Metode Gauss-Newton dengan Conjugate Gradient pada Masalah Klasifikasi Biner Menggunakan Regresi Logistik**. Skripsi. Program Studi Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Dr. Mohammad Jamhuri M.Si. (2) Erna Herawati M.pd.

Kata Kunci: Regresi Logistik, Gauss-Newton, Conjugate Gradient, Klasifikasi Biner, Optimasi

Klasifikasi biner merupakan permasalahan penting dalam pembelajaran mesin yang sering diselesaikan menggunakan regresi logistik. Namun, tantangan utama dari metode ini adalah proses optimasi parameter, terutama ketika dihadapkan pada data berdimensi besar. Penelitian ini mengusulkan penggabungan metode Gauss-Newton dan Conjugate Gradient untuk meningkatkan efisiensi dan akurasi proses pelatihan model. Dataset Ionosphere dari UCI digunakan sebagai studi kasus. Model diuji melalui tahapan preprocessing, normalisasi, pelatihan, dan evaluasi menggunakan metrik klasifikasi. Hasil menunjukkan bahwa model gabungan mencapai akurasi 0.8629, precision 0.8293, recall 1, F1-score 0.9067, dan waktu 570 ms pada data uji. Selain itu, dibandingkan metode Gradient Descent dan Adam, pendekatan ini menunjukkan konvergensi yang lebih cepat dan stabil. Penelitian ini memberikan kontribusi dalam pengembangan metode klasifikasi biner yang lebih efisien dan akurat.

## ABSTRACT

Sari, Silvi Puspita. 2025. **Performance Improvement of Gauss Newton Method with conjugate gradient on Binary Classification Problem Using Logistic Regression.** Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Maulana Malik Ibrahim Malang. Advisors: (1) Dr. Mohammad Jamhuri M.Si. (2) Erna Herawati M.pd.

Keywords: Logistic Regression, Gauss-Newton, Conjugate Gradient, Binary Classification, Optimization

Binary classification is an important problem in machine learning that is often solved using logistic regression. However, the main challenge of this method is the parameter optimisation process, especially when faced with large-dimensional data. This research proposes combining the Gauss-Newton and Conjugate Gradient methods to improve the efficiency and accuracy of the model training process. The Ionosphere dataset from UCI is used as a case study. The model was tested through the stages of preprocessing, normalisation, training, and evaluation using classification metrics. The results showed that the combined model achieved 0.8679 accuracy, 0.8293 precision, 1 recall, 0.9067 F1-score, and time of 570 ms on the test data. In addition, compared to Gradient Descent and Adam's method, this approach shows faster and more stable convergence. This research contributes to the development of more efficient and accurate binary classification methods.

## مستخلص البحث

ساري، سيلفي بوسيتا. 2025 - تحسين أداء طريقة غاوس نيوتن مع التدرج المترافق على مشكلة التصنيف الثنائي باستخدام الانحدار اللوجستي. أطروحة. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانغ. المشرفة: د. محمد جمهوري م.س. (2) إرنا هيراواتي م.د.

الكلمات المفتاحية: الانحدار اللوجستي، غاوس-نيوتن، التدرج المترافق، التصنيف الثنائي، التحسين.

عد التصنيف الثنائي مشكلة مهمة في التعلم الآلي يتم حلها غالبا باستخدام الانحدار اللوجستي. ومع ذلك ، فإن التحدي الرئيسي لهذه الطريقة هو عملية تحسين المعلمات ، خاصة عند مواجهة بيانات كبيرة الأبعاد. تقترح والتدرج المترافق لتحسين كفاءة ودقة عملية التدريب *Gauss-Newton* هذه الدراسة دمج طريقة كدراسة حالة. يتم اختبار النموذج من خلال *UCI* النموذجية. تم استخدام مجموعة بيانات الأيونوسفير من مراحل المعالجة المسبقة والتطبيع والتدريب والتقييم باستخدام مقاييس التصنيف. أظهرت النتائج أن النموذج *F1* 0.9067 حقق دقة قدرها 0.8629 ، ودقة 0.8293 ، واستدعاء 1 ، ودرجة 570 ، ووقت *Adam* و *Gradient Descent* مللي ثانية على بيانات الاختبار. بالإضافة إلى ذلك ، بالمقارنة مع طرق ، يظهر هذا النهج تقاربا أسرع وأكثر استقرارا. يساهم هذا البحث في تطوير طرق تصنيف ثنائية أكثر كفاءة ودقة.

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Klasifikasi biner merupakan masalah penting yang digunakan untuk menganalisis data terutama dalam bidang ilmu terapan seperti data mining, pembelajaran mesin, dan statistika. Dengan perkembangan teknologi dan meningkatnya jumlah data, tantangan yang dihadapi dalam klasifikasi biner menjadi semakin kompleks (Hastie et al., 2009). Metode utama yang sering digunakan untuk menangani klasifikasi biner adalah regresi logistik. Regresi logistik memiliki arti yang luas, seperti memprediksi penyakit dalam bidang biomedis, mengelompokkan pelanggan dalam ilmu ekonomi (Hosmer, 2013), hingga melakukan analisis di bidang sosial media (Murphy, 2012). Keefektifannya dalam berbagai bidang menunjukkan pentingnya metode ini sebagai alat utama dalam menangani masalah klasifikasi biner.

Regresi logistik bekerja dengan memanfaatkan fungsi sigmoid (logistik) untuk memprediksi probabilitas suatu kejadian berdasarkan nilai variabel inputnya (Bishop, 2006). Namun, meskipun efektif metode ini menghadapi tantangan signifikan, terutama ketika diterapkan pada data berskala besar. Salah satu tantangan utama adalah proses optimasi yang diperlukan untuk menemukan solusi terbaik dalam meminimalkan fungsi kesalahan atau memaksimalkan fungsi *likelihood* (Lan, 2004). Dalam konteks data berskala besar, proses optimasi menjadi semakin krusial karena secara langsung mempengaruhi performa regresi logistik. Oleh karena itu, pengembangan metode optimasi yang lebih efektif menjadi

kebutuhan penting untuk meningkatkan kemampuan regresi logistik dalam menangani tantangan klasifikasi biner yang semakin kompleks.

Penelitian terdahulu oleh (Afifah, 2020; Umaroh, 2020) menggunakan metode Newton-Raphson untuk optimasi pada model regresi logistik, khususnya pada data kesejahteraan rumah tangga. Sedangkan (Sutrisno, 2020) menggunakan pendekatan serupa pada data ibu hamil yang menderita anemia. Meskipun pendekatan Newton-Raphson mampu menyelesaikan masalah optimasi, penelitian ini menunjukkan adanya keterbatasan dalam hal efisiensi, terutama ketika diterapkan pada data besar.

Untuk mengatasi keterbatasan metode Newton-Raphson, penelitian ini mengusulkan pendekatan baru yang menggabungkan metode optimasi Gauss-Newton dan conjugate gradient. Metode Gauss-Newton efektif untuk masalah optimasi non-linear seperti regresi logistik, namun terbukti kurang efisien dengan data yang berdimensi besar (Nocedal & Wright, 2006). Sebaliknya, metode conjugate gradient tidak memerlukan penyimpanan matriks Hessian, sehingga lebih cocok untuk data berskala besar dan memungkinkan iterasi yang lebih cepat (Dembo et al., 1982). Penggabungan kedua metode ini diharapkan dapat mengatasi keterbatasan masing-masing, meningkatkan performa model, dan memberikan efisiensi yang lebih baik dalam klasifikasi biner pada data berukuran besar.

Pada penelitian ini metode Gauss-Newton dan conjugate gradient akan diimplementasikan pada data *Ionosphere*. Penggunaan dua dataset ini bertujuan untuk menguji keandalan pendekatan optimasi yang diusulkan dalam kasus nyata. Penelitian ini bertujuan untuk meningkatkan akurasi dan efisiensi prediksi dalam klasifikasi biner melalui metode optimasi yang diusulkan. Penelitian ini diharapkan

dapat memberikan kontribusi yang signifikan dalam pengembangan teknik klasifikasi yang lebih efisien. Kebaruan dari penelitian yang diusulkan ini terletak pada penggunaan Metode Gauss-Newton beserta conjugate gradient untuk klasifikasi biner menggunakan regresi logistik yang belum pernah dilakukan pada penelitian-penelitian sebelumnya.

Dalam perspektif islam tantangan dalam pengelolaan data, termasuk regresi logistik tidak hanya terbatas pada aspek teknis, tetapi harus mencakup keakuratan dan kejujuran dalam setiap tahapnya. Data yang digunakan harus mencerminkan realitas, dan proses analisis harus dilakukan dengan prinsip keadilan dan tanggung jawab. Prinsip ini sesuai dengan nilai-nilai islam yang menekankan pentingnya kebenaran dan kejujuran, sebagaimana tercermin dalam surat Al-Mursalat ayat 35:

هَذَا يَوْمٌ لَا يَنْطِقُونَ ﴿٣٥﴾

(Kementerian Agama, 2022)

Artinya: “Ini adalah hari keputusan, kami mengumpulkan kamu dengan orang-orang terdahulu.” (QS. Al Mursalat [77]: 35)

Ayat ini menggambarkan pentingnya pengumpulan data secara komprehensif untuk memastikan keadilan. Dalam konteks pengolahan data, dapat dimaknai sebagai keharusan untuk menghimpun data secara lengkap tanpa manipulasi, sehingga keputusan atau hasil analisis yang dihasilkan dapat mencerminkan kondisi yang sebenarnya. Selain itu, Allah SWT dalam surat Al-Ghasiyah ayat 25-26:

إِنَّ إِلَيْنَا إِيَابَهُمْ ﴿٢٥﴾ ثُمَّ إِنَّ عَلَيْنَا حِسَابَهُمْ ﴿٢٦﴾

(Kementerian Agama, 2022)

Artinya: “Sesungguhnya kepada Kami lah mereka kembali. Kemudian, sesungguhnya kewajiban Kami lah untuk menghisap mereka.” (QS. Al-Ghasiyah [88]: 25-26).

Ayat ini menekankan bahwa setiap manusia akan dihisab oleh Allah SWT berdasarkan fakta yang objektif. Tidak diperlukan saksi dari manusia karena anggota tubuh seperti mata, tangan, dan kaki akan bersaksi atas segala perbuatan mereka. Sebagaimana surat Yassin ayat 65:

الْيَوْمَ نَخْتِمُ عَلَىٰ أَفْوَاهِهِمْ وَتُكَلِّمُنَا أَيْدِيهِمْ وَتَشْهَدُ أَرْجُلُهُمْ بِمَا كَانُوا يَكْسِبُونَ ﴿٦٥﴾

(Kementerian Agama, 2022)

Artinya: “Pada hari itu Kami akan membungkam mulut mereka sehingga mereka tidak dapat berdusta atau beralasan apapun. Saat itu, tangan mereka akan berbicara kepada Kami tentang apa yang telah diperbuat selama di dunia, dan kaki mereka akan menjadi saksi atas segala perbuatan yang telah mereka lakukan. Dengan demikian, mereka tidak akan mampu menghindar dari dosa-dosa yang telah diperbuat.” (QS. Yassin [36]: 65)

Dalam analisis data, pentingnya menjaga kejujuran, karena data yang diproses merupakan saksi digital yang akan memberikan gambaran realitas. Manipulasi data adalah bentuk ketidakjujuran yang tidak hanya merusak hasil analisis, tetapi juga bertentangan dengan nilai-nilai islam. Namun, proses menganalisis data harus dilakukan dengan menjaga integritas moral, dimana setiap langkah pengolahan data dilakukan berdasarkan prinsi Al-Haqq (kebenaran) dan Al-Adl (keadilan) yang di ajarkan dalam islam.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana penerapan conjugate gradient pada metode Gauss-Newton dalam pelatihan model regresi logistik untuk klasifikasi biner?
2. Bagaimana metode conjugate gradient dapat meningkatkan akurasi dan efisiensi prediksi dalam klasifikasi biner dengan metode Gauss-Newton?

### 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah sebagai berikut:

1. Menganalisis penerapan metode conjugate gradient pada metode Gauss-Newton dalam pelatihan model regresi logistik untuk klasifikasi biner.
2. Mengevaluasi sejauh mana metode conjugate gradient dapat meningkatkan akurasi dan efisiensi prediksi dalam klasifikasi biner menggunakan metode Gauss-Newton.

### 1.4 Manfaat Penelitian

Manfaat yang diharapkan dalam penelitian ini yaitu: (manfaat teoritis dan praktis)

#### 1. Manfaat Teoritis

Penelitian ini memberikan manfaat teoritis sebagai berikut:

- a. Menyediakan kontribusi dalam pengembangan metode optimasi dalam regresi logistik dengan menggabungkan metode Gauss-Newton dan conjugate gradient, yang dapat memperluas wawasan tentang algoritma optimasi.
- b. Hasil penelitian ini dapat menjadi dasar atau rujukan untuk penelitian lain yang fokus pada pengembangan algoritma optimasi atau aplikasi regresi logistik dalam bidang lain.
- c. Menghasilkan pendekatan gabungan baru yang dapat diuji keandalannya secara matematis dan aplikasinya dalam berbagai masalah klasifikasi biner.

- d. Memperdalam pemahaman tentang efisiensi dan keterbatasan metode Gauss-Newton dan conjugate gradient dalam konteks data besar.

## 2. Manfaat Praktis

Penelitian ini memberikan manfaat praktis sebagai berikut:

- a. Peningkatan akurasi prediksi dapat memungkinkan pengguna model regresi logistik untuk mencapai tingkat akurasi prediksi yang lebih baik dalam pengambilan keputusan berbasis data.
- b. Penghematan sumber daya, dengan iterasi yang lebih cepat dan penggunaan memori yang lebih efisien, pendekatan ini dapat mengurangi kebutuhan sumber daya komputasi, yang penting dalam pengolahan data besar.
- c. Relevansi dalam dunia nyata sehingga validasi pada dataset seperti *Ionosphere* menunjukkan relevansi dan keandalan metode ini dalam menangani masalah nyata di berbagai domain.

### 1.5 Batasan Masalah

Untuk mendekati sasaran yang diharapkan, maka perlu diadakan pembatasan permasalahan dalam penelitian ini yaitu sebagai berikut:

1. Penelitian ini hanya dibatasi oleh regresi logistik untuk klasifikasi biner, sehingga tidak mencakup regresi logistik lainnya.
2. Penelitian ini hanya fokus pada metode Gauss-Newton dan conjugate gradient, sehingga hasil penelitian tidak berlaku untuk metode optimasi lain.

## 1.6 Definisi Istilah

### 1. Regresi Logistik

Regresi Logistik adalah metode statistik yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen dengan variabel dependen yang bersifat biner (dua kategori). Model ini menghitung probabilitas suatu kejadian dengan menggunakan fungsi logit untuk menangkap hubungan nonlinear antara variabel-variabel tersebut (Scott et al., 1991).

### 2. Klasifikasi Biner

Klasifikasi Biner adalah proses membagi data menjadi dua kategori berdasarkan karakteristik tertentu. Dalam konteks regresi logistik, klasifikasi biner digunakan untuk memprediksi dua kemungkinan hasil, seperti “ya” atau “tidak,” “benar” atau “salah,” atau “positif” dan “negatif” (Bishop, 2006).

### 3. Metode Gauss-Newton

Metode Gauss-Newton adalah algoritma iteratif yang digunakan untuk meminimalkan jumlah kuadrat kesalahan (*least squares*) pada model nonlinear. Metode ini menggabungkan pendekatan turunan pertama untuk memperbaiki estimasi parameter hingga mendekati solusi optimal (Nocedal & Wright, 2006).

### 4. Metode conjugate gradient

Metode conjugate gradient adalah teknik numerik iteratif yang digunakan untuk menyelesaikan sistem persamaan linear atau untuk meminimalkan fungsi kuadrat. Metode ini sangat efisien dalam menangani masalah berdimensi besar dan matriks jarang (Hestenes & Stiefel, 1952).

## 5. Probabilitas

Probabilitas adalah ukuran kuantitatif dari kemungkinan terjadinya suatu peristiwa. Dalam regresi logistik, probabilitas dinyatakan dalam fungsi sigmoid untuk memprediksi hasil biner berdasarkan nilai variabel independent (Neuhaus & Mcculloch, 2010).

## 6. Parameter Model

Parameter model adalah koefisien dalam regresi logistik yang mempresentasikan pengaruh variabel independen terhadap variabel dependen, nilai parameter ini ditentukan melalui proses estimasi yang mengoptimalkan kecocokan model dengan data (Barnett et al., 2005)

## 7. Iterasi

Iterasi adalah proses pengulangan langkah-langkah tertentu hingga mencapai hasil yang diinginkan. Dalam metode Gauss-Newton dan conjugate gradient, iterasi digunakan untuk mendekati solusi optimal dengan meningkatkan nilai parameter model di setiap langkah (Shewchuk, 1994)

## 8. Optimasi

Optimasi adalah proses menemukan nilai terbaik dari suatu fungsi objektif dalam kondisi tertentu. Dalam penelitian ini, optimasi digunakan untuk menentukan parameter model yang meminimalkan kesalahan prediksi pada regresi logistik (Nocedal & Wright, 2006).

## BAB II TINJAUAN PUSTAKA

### 2.1 Klasifikasi Biner

Klasifikasi biner merupakan jenis masalah klasifikasi yang bertujuan untuk mengklasifikasi data kedalam dua kategori atau kelas yang saling eksklusif. Masalah ini adalah bentuk spesifik dari masalah klasifikasi, dimana hanya ada dua kelas atau label yang mungkin. Klasifikasi biner sering digunakan dalam berbagai aplikasi, seperti deteksi penipuan, diagnosa medis, pengenalan wajah maupun pembelajaran mesin (Bishop, 2006).

Pada dasarnya, klasifikasi biner dapat dipandang sebagai masalah prediksi yang memetakan data masukan (fitur) kedalam dua kelas yang saling berlawanan atau berbeda. Misalnya, dalam konteks pengenalan email, sistem harus memutuskan apakah sebuah email termasuk kategori spam atau tidak spam. Begitu juga dalam diagnose medis, sistem dapat memutuskan apakah seorang pasien menderita penyakit tertentu atau tidak.

Klasifikasi biner dapat dijelaskan melalui pendekatan matematis sebagai berikut: diberikan sebuah dataset yang terdiri dari pasangan input berupa vektor fitur dan output berupa label kelas. Tujuan utama dari klasifikasi biner adalah untuk menemukan fungsi yang memetakan vektor fitur ke dalam salah satu atau dua kelas. Fungsi ini dapat dinyatakan sebagai berikut:

$$f: \mathbb{R}^n \rightarrow \{0,1\}$$

di mana  $\mathbb{R}^n$  adalah ruang fitur berdimensi  $n$ , dan  $\{0,1\}$  adalah ruang output yang menunjukkan dua kelas (label biner).

Klasifikasi biner menjadi krusial dalam pembelajaran mesin, karena banyak aplikasi di dunia nyata yang dapat dikategorikan dalam dua kelas yang jelas. Misalnya, pada aplikasi deteksi penyakit, tujuan utamanya adalah untuk memprediksi apakah seorang pasien memiliki penyakit atau tidak. Pada aplikasi pemasaran, sebuah sistem berusaha untuk mengidentifikasi apakah seorang konsumen akan membeli produk atau tidak berdasarkan data perilaku mereka. Dalam konteks pembelajaran mesin, klasifikasi biner sering dijadikan sebagai masalah pembelajaran (*supervised learning*). Pada model pembelajaran terawasi, model dilatih menggunakan data pelatihan yang sudah dilabeli dengan benar. Proses pelatihan melibatkan pencarian pola atau hubungan yang dapat digunakan untuk memprediksi label kelas pada data yang belum dilabeli.

Klasifikasi biner merupakan metode yang banyak diterapkan dalam berbagai bidang dan industri. Dalam bidang keuangan, sistem klasifikasi biner digunakan untuk mendeteksi transaksi yang mencurigakan atau penipuan, dengan mengklasifikasikan setiap transaksi penipuan atau bukan penipuan, berdasarkan fitur seperti lokasi, jumlah transaksi, dan juga histori penggunaan sebelumnya (Prasad et al., 2009). Dalam dunia medis, klasifikasi biner membantu dalam diagnosa kesehatan pasien, misalnya menentukan apakah seorang pasien mengidap kanker berdasarkan hasil pemeriksaan medis (Zou, H., & Hastie, 2005). Pada pengelompokan dokumen, klasifikasi biner dimanfaatkan dalam pemfilteran email untuk membedakan antara spam dan bukan spam (Pang, B., & Lee, 2008). Pada bidang teknologi, sistem pengenalan wajah dan objek menggunakan klasifikasi biner untuk mengidentifikasi apakah wajah dalam gambar dikenali atau tidak, atau menentukan apakah suatu objek termasuk kategori tertentu (Kouh, et al., 2008).

Selain itu, sistem rekomendasi juga sering menggunakan klasifikasi biner untuk memprediksi preferensi pengguna, seperti memutuskan apakah seorang pengguna akan menyukai atau tidak menyukai suatu produk berdasarkan riwayat interaksi mereka sebelumnya (Rendle, 2012).

## 2.2 Regresi Logistik

Regresi logistik adalah model probabilistik yang digunakan untuk klasifikasi biner, yaitu memprediksi keluaran  $y \in \{0,1\}$  berdasarkan variabel input  $\vec{x}$ . Tujuan utama dari regresi logistik biner adalah menemukan model terbaik, paling efektif, dan paling sesuai untuk menggambarkan hubungan antara variabel dependen dengan sekumpulan variabel independen. Regresi logistik mencakup berbagai jenis model, seperti regresi logistik dengan respon biner, ordinal, dan multinomial (Hosmer and Lemeshow, 1991).

Data yang digunakan dalam penelitian ini bisa dinyatakan kedalam bentuk matriks  $X$  dan  $Y$  sebagai berikut:

1. Matriks fitur:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

2. Matriks Label:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Selanjutnya matriks  $X$  juga bisa dinyatakan sebagai  $X = (\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_m)$

dengan

$$\vec{x}_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$$

Diberikan contoh pasangan data pelatihan  $(\vec{x}_i, y_i)$ ,  $\forall_i = 1, 2, \dots, m$  dengan  $\vec{x}_i \in \mathbb{R}^n$  dan  $y_i \in \mathbb{Z}$ . Dimana  $m$  adalah jumlah sampel (data) dalam dataset dan  $n$  adalah jumlah fitur (variabel) untuk setiap sampel.

Dari keterangan di atas, maka dapat didefinisikan:

### 1. Lapisan input

Lapisan input berfungsi sebagai tempat memasukkan data ke dalam model. Setiap sampel  $\vec{x}_i$  direpresentasikan sebagai vektor yang berisi nilai dari fitur-fitur (variabel) yang tersedia, di tambah dengan nilai konstan 1 untuk mempresentasikan bias.

$$\vec{x}_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$$

di mana:

$x_{i1}, x_{i2}, \dots, x_{in}$  adalah nilai fitur untuk sampel  $i$

Nilai konstan 1 pada awal vektor digunakan untuk mengalikan bobot bias  $\theta_0$

Lapisan ini menjadi penghubung antara dataset dan proses perhitungan dalam model

### 2. Koefisien bobot

Koefisien bobot  $\vec{\theta}$  adalah parameter yang dioptimalkan selama pelatihan model. Vektor bobot ini mempresentasikan hubungan atau kontribusi setiap fitur dalam menentukan output.

$$\vec{\theta} = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$$

di mana:

$\theta_0$  adalah bobot bias yang bekerja bersama nilai konstan 1.

$\theta_1, \theta_2, \dots, \theta_n$  adalah bobot untuk setiap fitur dalam input.

Selama proses pelatihan, nilai bobot diperbarui untuk meminimalkan kesalahan antara prediksi dan nilai aktual.

### 3. Fungsi Transfer

Fungsi transfer berfungsi untuk menggabungkan nilai input  $\vec{x}_i$  dengan bobot  $\vec{\theta}$  untuk menghasilkan nilai  $z_i$ , perhitungannya menggunakan operasi *dot product*:

$$z_i = \vec{x}_i \cdot \vec{\theta}$$

Secara eksplisit dapat ditulis sebagai:

$$z_i = \theta_0 + x_{i1}\theta_1 + x_{i2}\theta_2, \dots + x_{in}\theta_n$$

Nilai  $z_i$  hasil antara kontribusi setiap fitur dan bobotnya.

### 4. Output layer

Lapisan output bertugas mengubah nilai kombinasi linier  $z_i$  menjadi nilai probabilitas menggunakan fungsi sigmoid.

$$h_i = \frac{1}{1 + \exp(-z_i)} \quad (2.1)$$

di mana:

fungsi sigmoid memetakan nilai  $z_i$  ke dalam rentang (0,1) yang sesuai untuk interpretasi probabilitas.

Nilai  $h_i$  menunjukkan probabilitas bahwa sampel  $i$  termasuk dalam kelas positif ( $y_i = 1$ ).

### 5. Prediksi

Proses prediksi berfungsi untuk mengubah nilai probabilitas menjadi label kelas yang dapat digunakan untuk analisis atau pengambilan keputusan.

$$y_i = \begin{cases} 1, & h_i \geq 0.5 \\ 0, & h_i \leq 0.5 \end{cases} \quad (2.2)$$

di mana

$y_i$  adalah label kelas untuk sampel  $\vec{x}_i$ , dengan

$y_i = 1$ , menunjukkan kelas positif

$y_i = 0$ , menunjukkan kelas negatif

Menurut (Jamhuri et al., 2022) eror untuk satu data didefinisikan sebagai:

$$L_i = -(y_i \log(h_i) + (1 - y_i) \log(1 - h_i))$$

Fungsi Loss yang digunakan adalah *binary cross entropy* yang didefinisikan sebagai berikut:

$$L(\vec{\theta}) = \frac{1}{m} \sum_{i=1}^m L_i$$

atau

$$L(\vec{\theta}) = \frac{1}{m} \sum_{i=1}^m -(y_i \log(h_i) + (1 - y_i) \log(1 - h_i)) \quad (2.3)$$

Proses pelatihan model melibatkan *feedforward*, *loss computation*, *backward*, dan prediksi, dengan penjelasan sebagai berikut.

1. *Feedforward*:

Probabilitas  $h$  dihitung untuk setiap data dalam dataset. Matriks data input  $X \in \mathbb{R}^{m \times n}$ , dengan  $m$  adalah jumlah data. Menghasilkan probabilitas  $h$  merujuk pada persamaan (2.1).

2. *Loss computation*

Fungsi loss yang digunakan adalah *log-likelihood*, yang sudah dinyatakan dalam persamaan (2.3).

### 3. *Backward* (Optimasi)

Proses *backward* melibatkan perhitungan gradient  $\nabla L(\vec{\theta})$  untuk meminimalkan  $L(\vec{\theta})$ . Gradiennya dinyatakan sebagai:

$$\nabla L(\vec{\theta}) = \left( \frac{\partial L}{\partial \theta_0}, \frac{\partial L}{\partial \theta_1}, \frac{\partial L}{\partial \theta_2}, \dots, \frac{\partial L}{\partial \theta_n} \right)$$

Untuk menentukan fungsi biaya pada klasifikasi biner, bisa menggunakan persamaan (2.3). Kemudian untuk meminimalkan fungsi biaya bisa menggunakan *gradient descent* sebagai berikut:

$$\theta_j^{(k+1)} = \theta_j^{(k)} - \eta \frac{\partial L(\vec{\theta})}{\partial \theta_j} \quad (2.4)$$

di mana

$$\frac{\partial L(\vec{\theta})}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m y_i \frac{\partial \log(h_i)}{\partial \theta_j} + (1 - y_i) \frac{\partial \log(1 - h_i)}{\partial \theta_j} \quad (2.5)$$

untuk  $j = 0, 1, \dots, n$

Kemudian dimisalkan  $b = \log h_i$  dan  $c = \log(1 - h_i)$ , maka turunan dari  $b$  dan  $c$  terhadap  $\theta_j$  adalah sebagai berikut:

Turunan  $\log(h_i)$  terhadap  $\theta_j$  adalah

$$\frac{\partial b}{\partial \theta_j} = \frac{\partial \log(h_i)}{\partial h_i} \frac{\partial h_i}{\partial z_i} \frac{\partial z_i}{\partial \theta_j} = \frac{1}{h_i} h_i (1 - h_i) x_{ij} = (1 - h_i) x_{ij} \quad (2.6)$$

Turunan  $\log(1 - h_i)$  terhadap  $\theta_j$ :

$$\begin{aligned} \frac{\partial c}{\partial \theta_j} &= \frac{\partial \log(1 - h_i)}{\partial (1 - h_i)} \frac{\partial (1 - h_i)}{\partial z_i} \frac{\partial z_i}{\partial \theta_j} \\ &= \frac{1}{1 - h_i} (-h_i (1 - h_i)) x_{ij} \\ &= -h_i x_{ij} \end{aligned} \quad (2.7)$$

Kemudian substitusi  $\frac{\partial b}{\partial \theta_j}$  dan  $\frac{\partial c}{\partial \theta_j}$  ke persamaan (2.7) dan (2.8) sehingga

menjadi

$$\frac{\partial L(\vec{\theta})}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (y_i - h_i) x_{ij}, \quad \forall j = 0, 1, \dots, n \quad (2.8)$$

Untuk update parameter  $\theta_j$ , menggunakan persamaan (2.8) ke persamaan

(2.4) maka akan menghasilkan

$$\theta_j^{(k+1)} = \theta_j^{(k)} + \eta \cdot \frac{1}{m} \sum_{i=1}^m (y_i - h_i) x_{ij}, \quad \forall j = 0, 1, \dots, n \quad (2.9)$$

di mana:

$\eta = \text{Learning rate}$

$h_i = \text{Fungsi sigmoid}$

#### 4. Prediksi

Model mengklasifikasikan keluaran  $y$  berdasarkan probabilitas pada persamaan (2.2). Regresi logistik umum digunakan untuk berbagai masalah klasifikasi biner, namun memerlukan optimasi yang efisien untuk dataset besar.

Dalam proses optimasi *log-likelihood*, metode Gauss-Newton sering digunakan untuk menangani sistem nonlinier yang muncul dari turunan fungsi *log-likelihood*. Sementara itu, metode conjugate gradient merupakan alternatif yang lebih efisien untuk dataset besar karena dapat mengurangi kebutuhan memori dan komputasi tanpa menghitung matriks Hessian secara eksplisit (Fletcher, 1987). Pendekatan gabungan kedua metode ini menawarkan kecepatan konvergensi yang tinggi (karakteristik Gauss-Newton) sambil menjaga efisiensi komputasi

(karakteristik conjugate gradient). Implementasi ini sangat relevan untuk masalah klasifikasi biner berskala besar.

### 2.3 Metode Gauss-Newton

Metode Gauss-Newton adalah salah satu algoritma optimasi yang digunakan untuk memecahkan masalah *nonlinier least squares*, yaitu Ketika fungsi objektif yang ingin diminimalkan merupakan jumlah kuadrat dari residual yang bersifat nonlinier terhadap parameter. Metode ini merupakan penyederhanaan metode *Newton-Raphson* dengan mengabaikan komponen turunan kedua (Hessian), yang membuatnya lebih efisien dalam praktik, terutama untuk masalah dengan dimensi besar (Nocedal & Wright, 2006). yang disederhanakan untuk kasus khusus di mana fungsi objektifnya adalah jumlah kuadrat galat. Dengan demikian, metode ini sering digunakan dalam regresi nonlinier dan masalah optimasi kuadrat kecil (Bjorck, 1996).

Tujuan utama metode ini adalah meminimalkan fungsi loss yang dalam konteks regresi logistik pada persamaan berikut

$$L_i = -(y_i \log h_i + (1 - y_i) \log (1 - h_i)) \quad (2.10)$$

dalam penelitian ini, dilakukan linearisasi  $\log h_i$  dan  $\log(1 - h_i)$  dalam persamaan  $L_i$  menggunakan ekspansi deret Taylor orde pertama. Dapat dimisalkan  $b(\vec{\theta}) = \log(h_i)$  dan  $c(\vec{\theta}) = (1 - h_i)$ , maka ekspansi deret untuk fungsi-fungsi ini adalah

$$b(\vec{\theta} + \Delta\vec{\theta}) \approx b(\vec{\theta}) + \nabla b(\vec{\theta}) \cdot (\Delta\vec{\theta}) \quad (2.11)$$

$$c(\vec{\theta} + \Delta\vec{\theta}) \approx c(\vec{\theta}) + \nabla c(\vec{\theta}) \cdot (\Delta\vec{\theta})$$

Gradien dari  $b$  dan  $c$  dapat dituliskan sebagai berikut

$$\nabla b(\vec{\theta}) = \left( \frac{\partial b}{\partial \theta_0}, \frac{\partial b}{\partial \theta_1}, \dots, \frac{\partial b}{\partial \theta_n} \right), \quad \nabla c(\vec{\theta}) = \left( \frac{\partial c}{\partial \theta_0}, \frac{\partial c}{\partial \theta_1}, \dots, \frac{\partial c}{\partial \theta_n} \right) \quad (2.12)$$

$$\Delta \vec{\theta} = (\Delta \vec{\theta}_0, \Delta \vec{\theta}_1, \dots, \Delta \vec{\theta}_n)$$

Turunan dari  $b$  dan  $c$  terhadap  $\theta_j$ , sebagaimana yang telah didefinisikan dalam persamaan (2.6) dan (2.7) adalah

$$\frac{\partial b}{\partial \theta_j} = (1 - h_i)x_{ij}, \quad \frac{\partial c}{\partial \theta_j} = -h_i x_{ij} \quad (2.13)$$

$$\forall j = 0, 1, 2, 3, \dots, n$$

Untuk melinierisasi persamaan (2.10) dengan menggunakan fungsi aproksimasi  $b$  dan  $c$  pada persamaan (2.12) dan turunannya pada persamaan (2.13), sehingga

$$L_i = -y_i \left( b(\vec{\theta}) + \nabla b(\vec{\theta}) \cdot (\Delta \vec{\theta}) \right) - (1 - y_i) c(\vec{\theta}) + \nabla c(\vec{\theta}) \cdot (\Delta \vec{\theta})$$

Persamaan di atas dapat di tulis sebagai

$$y_i (\nabla b(\vec{\theta}) \cdot \Delta \vec{\theta}) + (1 - y_i) (\nabla c(\vec{\theta}) \cdot \Delta \vec{\theta}) = -y_i b(\vec{\theta}) - (1 - y_i) c(\vec{\theta})$$

Dari persamaan di atas, dapat ditulis bahwa bagian yang tidak diketahui diberi nama *left hand side* dan bagian yang diketahui diberi nama *right hand side*. Dengan mengelompokkan  $\Delta \vec{\theta}$  ke bagian *left hand side* maka akan memperoleh

$$\left( y_i \nabla b(\vec{\theta}) - y_i \nabla c(\vec{\theta}) + \nabla c(\vec{\theta}) \right) \cdot \Delta \vec{\theta} = - \left( y_i b(\vec{\theta}) + (1 - y_i) c(\vec{\theta}) \right) \quad (2.14)$$

Selanjutnya, selisih antara  $\nabla b(\vec{\theta})$  dan  $\nabla c(\vec{\theta})$ , yaitu  $\nabla b(\vec{\theta}) - \nabla c(\vec{\theta})$  dapat dimisalkan sebagai sebuah vektor yang terdiri dari elemen-elemen  $(1, x_{i1}, x_{i2}, \dots, x_{in})$ . Dalam konteks ini, elemen pertama vektor tersebut yaitu 1, biasanya mempresentasikan komponen bias atau konstanta, sementara elemen-elemen berikutnya, yaitu  $(x_{i1}, x_{i2}, \dots, x_{in})$ , merupakan fitur-fitur atau variabel data

yang relevan untuk obeservasi ke- $i$ . Vektor ini secara keseluruhan diberi nama  $\vec{x}_i$ , yang bertindak sebagai representasi sederhana.

$$y_i (\nabla b(\vec{\theta}) - y_i \nabla c(\vec{\theta})) = y_i \vec{x}_i \quad (2.15)$$

kemudian substitusikan persamaan (2.15) ke dalam persamaan (2.14) ke bagian *left hand side* dan menyederhanakan *right hand side* dengan fungsi *cost function*.

$$y_i (\nabla b(\vec{\theta}) - \nabla c(\vec{\theta})) + \nabla c(\vec{\theta}) = y_i \vec{x}_i - (h_i) \vec{x}_i = (y_i - h_i) \vec{x}_i \quad (2.16)$$

Jika persamaan (2.16) di masukkan ke dalam persamaan (2.14) maka didefinisikan

$$(y_i - h_i) \vec{x}_i \cdot \Delta \vec{\theta} = -(y_i \log h_i + (1 - y_i) \log(1 - h_i)) \quad (2.17)$$

dengan substitusi  $i$  ke dalam persamaan (2.17) untuk  $i = 1, 2, \dots, m$ , maka akan memperoleh himpunan persamaan sistem linier  $m$  di bawah ini

$$\begin{aligned} (y_1 - h_1) \vec{x}_1 \cdot \Delta \vec{\theta} &= L_1 \\ (y_2 - h_2) \vec{x}_2 \cdot \Delta \vec{\theta} &= L_2 \\ &\vdots \\ (y_m - h_m) \vec{x}_m \cdot \Delta \vec{\theta} &= L_m \end{aligned}$$

di mana  $L_i = -((y_i \log h_i) + (1 - y_i) \log(1 - h_i))$ . Sehingga diperoleh sistem persamaan linier dalam bentuk matriks  $J \Delta \theta = D$  dengan memisalkan

$$AJ = \begin{bmatrix} (y_1 - h_1) & (y_1 - h_1)x_{11} & \cdots & (y_1 - h_1)x_{1n} \\ (y_2 - h_2) & (y_2 - h_2)x_{21} & \cdots & (y_2 - h_2)x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ (y_m - h_m) & (y_m - h_m)x_{m1} & \cdots & (y_m - h_m)x_{mn} \end{bmatrix}, \quad (2.18)$$

$$W = \begin{bmatrix} \Delta \theta_0 \\ \Delta \theta_1 \\ \vdots \\ \Delta \theta_n \end{bmatrix}, \quad B = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{bmatrix}, \quad \Theta = \begin{bmatrix} \Delta \theta_0 \\ \Delta \theta_1 \\ \vdots \\ \Delta \theta_n \end{bmatrix}, \quad D = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_m \end{bmatrix}$$

Jika  $m = n$ , maka sistem memiliki satu solusi, jika  $m < n$  maka sistem memiliki banyak solusi karena terdapat kebebasan yang lebih besar, dan jika  $m > n$ , maka sistem tidak memiliki solusi eksak sehingga solusinya dapat diperoleh melalui pendekatan kuadrat terkecil (*Least Square Problem*), yang dinyatakan sebagai:

$$A^T A W = A^T B J^T J \Theta = J^T D \quad (2.19)$$

Misalkan  $A = J^T J$  dan  $B = J^T D$ , maka persamaan (2.19) dapat dituliskan menjadi

$$A \Theta = B \quad (2.20)$$

Karena

$$\Delta \theta_j = \theta_j^{k+1} - \theta_j^k, \quad \forall i = 0, 1, 2, \dots, n$$

Maka

$$\theta_j^{k+1} = \theta_j^k + \Delta \theta_j \quad (2.21)$$

dengan  $A \in \mathbb{R}^{n \times n}$ ,  $\Theta \in \mathbb{R}^{n \times 1}$ , dan  $B \in \mathbb{R}^{n \times 1}$ . Untuk memperbarui atau update bobot  $\vec{\theta}$ , dengan mengulang persamaan (2.19) dan (2.20) dengan toleransi error  $\Delta \theta_j \leq \varepsilon$  (Jamhuri et al., 2022). Selanjutnya persamaan (2.20) akan diselesaikan menggunakan *conjugate gradient* yang akan di bahas pada sub-bab berikutnya.

## 2.4 Metode Conjugate Gradient

Metode conjugate gradient adalah algoritma iteratif yang digunakan untuk menyelesaikan sistem persamaan linear simetris dan positif definit, atau untuk meminimalkan fungsi kuadrat dalam ruang berdimensi tinggi. Metode ini termasuk dalam kategori algoritma iteratif yang sangat efisien untuk masalah dalam ukuran besar, terutama ketika matriks koefisiennya jarang (Hestenes & Stiefel, 1952). Dari prespektif kalkulus multivariable, metode *conjugate gradient* bertujuan untuk meminimalkan fungsi kuadrat berikut

$$f(\theta) = \frac{1}{2}\theta^T A\theta - b^T \theta$$

Dengan gradien  $\nabla f(\theta) = A\theta - b$ .

Solusi optimum dicapai Ketika gradien bernilai nol, yaitu  $A\theta = b$ . Dalam banyak metode seperti Gradient Descent, arah pencarian selalu mengikuti arah negative gradien. Namun metode ini sering mengalami “zigzag” jika kontur fungsi berbentuk elips sempit. Conjugate gradient mengatasi kelemahan ini dengan memperhitungkan arah pencarian yang saling conjugate terhadap matriks  $A$ . Dua vektor arah  $\vec{d}_i$  dan  $\vec{d}_j$  disebut conjugate gradien  $A$  jika memenuhi

$$\vec{d}_i^T A \vec{d}_j = 0, \quad \forall i \neq j$$

Makna ini merupakan generalisasi dari ortogonalitas biasa (Euclidean), tetapi disesuaikan dengan struktur Matriks  $A$ . Keunggulan utama metode ini adalah efisiensi memori karena tidak perlu memerlukan penyimpanan matriks Hessian secara eksplisit (Nocedal & Wright, 2006).

Menurut (Sauer, 2017) ide-ide di balik metode conjugate gradient bergantung pada generalisasi konsep biasa dari *inner product*. *Euclidean inner product*  $(\vec{v}, \vec{w}) = \vec{v}^T \vec{w}$  bersifat simetris dan linier terhadap input  $\vec{v}$  dan  $\vec{w}$ , karena  $(\vec{v}, \vec{w}) = (\vec{w}, \vec{v})$  dan  $(\alpha \vec{u} + \beta \vec{w}, \vec{u}) = \alpha(\vec{v}, \vec{u}) + \beta(\vec{w}, \vec{u})$  untuk skalar  $\alpha$  dan  $\beta$ . *Euclidean inner product* juga bersifat positif-definit, yang berarti  $(\vec{v}, \vec{v}) > 0$  jika  $\vec{v} \neq 0$ .

Definisi 2.15 Misalkan  $A$  adalah matriks simetris positif-definit  $n \times n$ . Untuk dua  $n$   $\vec{v}$  dan  $\vec{w}$ , definisikan *A-inner product* sebagai:

$$(\vec{v}, \vec{w})_A = \vec{v}^T A \vec{w}$$

$\vec{v}$  dan  $\vec{w}$  adalah  $A$ -conjugate jika  $(\vec{v}, \vec{w})_A = 0$ . Dengan catatan bahwa *inner product* baru mewarisi sifat simetri, linearitas, dan positif definit dari matriks  $A$ . Karena  $A$  bersifat simetris, maka *inner product*  $A$  juga simetris:  $(\vec{v}, \vec{w})_A = \vec{v}^T A \vec{w} = (\vec{v}^T A \vec{w})^T = \vec{w}^T A \vec{v} = (\vec{w}, \vec{v})_A$ . *Inner product*  $A$  juga bersifat linier, dan sifat positif-definit berasal dari fakta bahwa jika  $A$  positif-definit, maka

$$(\vec{v}, \vec{v})_A = \vec{v}^T A \vec{v} > 0$$

Jika  $v \neq 0$ .

Secara teknis, metode conjugate gradient adalah metode langsung (*direct method*) dan menghasilkan solusi  $\Theta$  untuk sistem simetris positif-definit  $A\Theta = B$  dengan *finite loop* berikut:

$\Theta_0 =$  initial guess

$$\vec{d}_0 = \vec{r}_0 = B - A\Theta_0$$

for  $k = 0, 1, 2, \dots, n - 1$

If  $\vec{r}_k = 0$ , stop, end

$$\alpha_k = \frac{\vec{r}_k^T \vec{r}_k}{\vec{d}_k^T A \vec{d}_k} \tag{2.22}$$

$$\Theta_{k+1} = \Theta_k + \alpha_k \vec{d}_k \tag{2.2123}$$

$$\vec{r}_{k+1} = \vec{r}_k - \alpha_k A \vec{d}_k \tag{2.2224}$$

$$\beta_k = \frac{\vec{r}_{k+1}^T \vec{r}_{k+1}}{\vec{r}_k^T \vec{r}_k} \tag{2.25}$$

$$\vec{d}_{k+1} = \vec{r}_{k+1} + \beta_k \vec{d}_k \tag{2.26}$$

end

Metode conjugate gradient memperbarui tiga vektor berbeda di setiap langkah iterasi:

1.  $\Theta_k$ : Merupakan solusi aproksimasi pada langkah  $k$ . Nilai  $\Theta_k$  diperbarui setiap iterasi untuk mendekati solusi akhir sistem linier.
2.  $\vec{r}_k$  merupakan residu dari solusi aproksimasi  $\vec{\theta}_k$ . Residu ini didefinisikan sebagai:

$$\vec{r}_k = B - A\Theta_k$$

yang menunjukkan sejauh mana solusi aproksimasi  $\Theta_k$  belum memenuhi sistem  $A\Theta = B$ . Residu awal  $\vec{r}_0$  jelas didefinisikan sebagai  $\vec{r}_0 = B - A\Theta_0$ .

Selama iterasi:

$$J\Theta_{k+1} + \vec{r}_{k+1} = A(\Theta_k + \alpha_k \vec{d}_k) + \vec{r}_k - \alpha_k A\vec{d}_k = A\Theta_k + \vec{r}_k,$$

dan berdasarkan  $\vec{r}_k = \vec{b} - A\vec{\theta}_k$  untuk semua  $k$ , persamaan di atas memastikan bahwa residu diperbarui secara konsisten.

3.  $\vec{d}_k$ : Merupakan arah pencarian baru yang digunakan untuk memperbarui aproksimasi  $\Theta_k$  menjadi  $\Theta_{k+1}$ . Arah  $\vec{d}_k$  dihitung agar setiap langkah pencarian optimal terhadap residu sebelumnya.

Metode ini berhasil, karena setiap residu di atur agar ortogonal terhadap semua residu sebelumnya. Kunci utama mencapai ortogonalitas di antara residu adalah dengan memilih arah pencarian  $d_k$  yang saling *conjugate*. Konsep konjugasi merupakan perluasan dari ortogonalitas dan menjadi dasar nama algoritma. Sekarang dijelaskan tentang pemilihan  $\alpha_k$  dan  $\beta_k$ . Arah  $\vec{d}_k$  dipilih dari rentang ruang  $\vec{r}$  sebelumnya. Untuk memastikan bahwa residu berikutnya  $\vec{r}_{k+1}$  ortogonal terhadap semua residu sebelumnya, maka  $\alpha_k$  dipilih dengan tepat agar residu baru  $\vec{r}_{k+1}$  ortogonal terhadap arah  $d_k$ :

$$\Theta_{k+1} = \Theta_k + \alpha_k \vec{d}_k$$

$$\vec{b} - A\Theta_{k+1} = \vec{b} - A\Theta_k - \alpha_k A\vec{d}_k$$

$$\vec{r}_{k+1} = \vec{r}_k - \alpha_k A\vec{d}_k$$

$$0 = \vec{d}_k^T \vec{r}_{k+1} = \vec{d}_k^T \vec{r}_k - \alpha_k \vec{d}_k^T A\vec{d}_k$$

$$\alpha_k = \frac{\vec{d}_k^T \vec{r}_k}{\vec{d}_k^T A\vec{d}_k}$$

Karena  $\vec{d}_{k-1}$  ortogonal ke  $\vec{r}_k$ , maka:

$$\vec{d}_k - \vec{r}_k = \beta_{k-1} \vec{d}_{k-1}$$

$$\vec{r}_k^T \vec{d}_k - \vec{r}_k^T \vec{r}_k = 0$$

$$\vec{r}_k^T \vec{d}_k = \vec{r}_k^T \vec{r}_k$$

Selanjutnya, koefisien  $\beta_k$  dipilih untuk memastikan konjugasi  $A$  berpasangan

dengan  $\vec{d}_k$ :

$$\vec{d}_{k+1} = \vec{r}_{k+1} + \beta_k \vec{d}_k$$

$$0 = \vec{d}_k^T A\vec{d}_{k+1} = \vec{d}_k^T A\vec{r}_{k+1} + \beta_k \vec{d}_k^T A\vec{d}_k$$

$$\beta_k = -\frac{\vec{d}_k^T A\vec{r}_{k+1}}{\vec{d}_k^T A\vec{d}_k}$$

**Teorema 2.16** Misalkan  $A$  adalah matriks simetris positif-definit  $n \times n$  dan  $\vec{b} \neq 0$ .

Maka dalam conjugate gradient diasumsikan bahwa  $r_k \neq 0$  untuk  $k < n$  (jika  $r_k =$

0, maka persamaan sudah terselesaikan. Dengan syarat  $1 \leq k \leq n$ :

1. Tiga subruang berikut dari  $\mathbb{R}^n$  adalah sama:

$$\langle \vec{r}_1, \dots, \vec{x}_k \rangle = \langle \vec{r}_0, \dots, \vec{r}_{k-1} \rangle = \langle \vec{d}_0, \dots, \vec{r}_{k-1} \rangle$$

2. Residual  $\vec{r}_k$  saling ortogonal:  $\vec{r}_k^T \vec{r}_j = 0$  untuk  $j < k$

3. Arah  $\vec{d}_k$  saling conjugate terhadap matriks  $A$ :  $\vec{d}_k^T A\vec{d}_j = 0$  untuk  $j < k$

## Bukti

1. Untuk  $k = 1$ , perhatikan bahwa  $\langle \vec{x}_1 \rangle = \langle \vec{d}_0 \rangle = \langle \vec{r}_0 \rangle$ , karena  $\vec{x}_0 = 0$ .

Berdasarkan definisi  $\vec{x}_k = \vec{x}_{k+1} + \alpha_{k-1} \vec{d}_{k-1}$ . Menunjukkan bahwa dengan induksi  $\langle \vec{x}_1, \dots, \vec{x}_k \rangle = \langle \vec{d}_0, \dots, \vec{r}_{k-1} \rangle$ . Pendapat serupa dengan menggunakan  $\vec{d}_k = \vec{r}_k + \beta_{k-1} \vec{d}_{k-1}$  menunjukkan bahwa  $\langle \vec{r}_0, \dots, \vec{r}_{k-1} \rangle = \langle \vec{d}_0, \dots, \vec{r}_{k-1} \rangle$ .

2. Untuk (a) dan (b) lanjutkan dengan induksi. Ketika  $k = 0$ , tidak ada yang perlu dibuktikan. Asumsikan bahwa (a) dan (b) berlaku untuk  $k$ , dan akan dibuktikan bahwa (a) dan (b) juga berlaku untuk  $k + 1$  dengan  $\vec{r}_j^T$  di bagian kiri.

$$\vec{r}_j^T \vec{r}_{k+1} = \vec{r}_j^T \vec{r}_k - \frac{\vec{r}_k^T \vec{r}_k}{\vec{d}_k^T A \vec{d}_k} \vec{r}_j^T A \vec{d}_k \quad (2.2527)$$

Jika  $j < k - 1$ , maka  $\vec{r}_j^T \vec{r}_k = 0$  berdasarkan hipotesis induksi (b). Karena  $\vec{r}_j$  dapat dinyatakan sebagai kombinasi dari  $\vec{d}_0, \dots, \vec{d}_j$ , maka  $\vec{r}_j^T A \vec{d}_k = 0$  berdasarkan hipotesis induksi (c), dan (b) berlaku disisi lain, jika  $j = k$ , maka  $\vec{r}_k^T A \vec{d}_k + \beta_{k-1} \vec{d}_{k-1}^T A \vec{d}_k$ , menggunakan hipotesis induksi (c). pembuktian (b) selesai.

Karena  $\vec{r}_j^T \vec{r}_k = 0$ , persamaan (2.2526) dengan  $j < k + 1$  maka diberikan:

$$\frac{\vec{r}_{k+1}^T \vec{r}_{k+1}}{\vec{r}_k^T \vec{r}_k} = - \frac{\vec{r}_{k+1}^T A \vec{d}_k}{\vec{d}_k^T A \vec{d}_k}$$

dengan mengalikan definisi  $\vec{d}_{k+1}$  di sisi kiri dengan  $\vec{d}_j^T A$ , menghasilkan:

$$\vec{d}_k^T A \vec{d}_{k+1} - \frac{\vec{r}_{k+1}^T A \vec{d}_k}{\vec{d}_k^T A \vec{d}_k} \vec{d}_k^T A \vec{d}_k$$

Jika  $j = k$ , maka  $\vec{d}_k^T A \vec{d}_{k+1} = 0$  persamaan (2.2022) menggunakan simetri  $A$ .

Jika  $j \leq k - 1$ , maka  $A \vec{d}_j = \frac{\vec{r}_j - \vec{r}_{j+1}}{\alpha_j}$  (dari definisi  $\vec{r}_{k+1}$ ) adalah orthogonal

terhadap  $r_{k+1}$ , menunjukkan bahwa suku pertama pada sisi kanan persamaan

(2.2020) adalah nol, dan suku kedua juga nol berdasarkan hipotesis induksi.

Sehingga argument untuk (c) selesai.

## 2.5 Integrasi Al-Qur'an dalam Prespektif Islam

Perkembangan teknologi dan ilmu pengetahuan di era modern menghadirkan berbagai tantangan, salah satunya adalah pengolahan data secara akurat dan etis. Data yang akurat dan tidak manipulatif menjadi dasar pengambilan keputusan yang efektif dalam berbagai bidang, termasuk dalam klasifikasi biner menggunakan regresi logistik. Dalam pandangan islam, pengolahan data yang jujur dan bertanggung jawab adalah wujud implementasi nilai-nilai Al-Quran yang mengajarkan keadilan, kejujuran, dan akuntabilitas.

Dalam proses pengolahan data, prinsip kebenaran, kejujuran dan tanggung jawab moral menjadi tanggung jawab yang penting. Prinsip ini tidak hanya relevan secara etis, namun juga memiliki landasan spiritual yang kuat dalam ajaran islam. Al-Qur'an memberikan banyak panduan moral yang dapat menjadi pedoman dalam menjalankan tanggung jawab ini sesuai dengan Qs. Al-Ghafir ayat 17:

﴿١٧﴾ الْيَوْمَ تُجْزَىٰ كُلُّ نَفْسٍ بِمَا كَسَبَتْ ۗ فَلَا تُظْلَمُ يَوْمَ الْيَوْمِ ۗ فَلَمَّا لَآظَمَ الْيَوْمَ فَلَمَّا لَآظَمَ الْيَوْمَ سَرِيعَ الْحِسَابِ

(Kementerian Agama, 2022)

Artinya: “Pada hari itu, setiap jiwa akan diberi balasan sesuai dengan apa yang telah dikerjakannya. Tidak ada yang dirugikan sedikitpun. Sungguh Allah SWT sangat cepat perhitungannya.” (Q.S Al-Ghafir [40]: 17)

Ayat ini mengajarkan bahwa segala perbuatan manusia akan diperhitungkan secara adil dan tidak akan ada yang dirugikan. Dalam konteks analisis data ini menjadi sebuah peringatan untuk memastikan bahwa proses pengolahan data dilakukan dengan integritas, tanpa manipulasi. Data yang akurat dan transparan merupakan cerminan dari keadilan yang dituntut dalam islam. Proses ini seperti (hisab) atau perhitungan di akhirat, dimana tidak akan ada yang dirugikan Karena keadilan Allah SWT bersifat mutlak.

Sebagaimana manusia bertanggung jawab atas setiap perbuatannya, seorang peneliti data bertanggung jawab atas hasil analisis yang dihasilkan. Jika data tidak diolah secara akurat, hasil yang diperoleh akan merugikan pihak-pihak tertentu, baik dalam pengambilan keputusan maupun dampak yang ditimbulkan. Oleh karena itu, keakuratan data adalah bagian dari tanggung jawab moral yang sejalan dengan prinsip keadilan dalam islam.

Selain itu, ayat ini mengingatkan bahwa Allah SWT Maha Cepat dalam perhitungan, yang menjadi pengingat bahwa tidak ada sesuatupun yang luput dari pengawasan-Nya. Prinsip ini menegaskan pentingnya transparansi dalam setiap langkah pengolahan data untuk memastikan hasil yang sesuai dengan realitas dan tidak menimbulkan kerugian. Kemudian Allah SWT berfirman pada surat Ibrahim ayat 14:

وَلَنُصَلِّنَّكُمْ الْأَرْضَ مِنْ بَعْدِهِمْ فَلْيَذِكْ لِمَنْ خَافَ مَقَامِي وَخَافَ وَعِيدِ (١٤)

(Kementerian Agama, 2022)

Artinya: “Dan sungguh, Kami akan menempatkan kamu di negeri-negeri itu setelah mereka. Itu bagi orang yang takut (akan saat menghadap) kepada-Ku dan takut akan ancaman-Ku.” (Q.S Ibrahim [14]: 14)

Ayat ini menekankan pentingnya memiliki rasa takut kepada Allah SWT sebagai pendorong untuk selalu bertindak jujur dan adil. Dalam proses pengolahan

data, rasa takut kepada Allah SWT menjadi pengingat bahwa setiap langkah yang kita ambil, baik dalam pengumpulan, analisis, maupun pelaporan data, harus dilakukan dengan kejujuran dan tanpa penyimpangan.

Dalam konteks pengolahan data, ketakwaan tercermin dalam upaya untuk memastikan bahwa data yang digunakan dan hasilnya tidak dimanipulasi demi kepentingan pribadi ataupun kelompok. Seperti janji Allah SWT kepada orang-orang yang bertakwa, keberhasilan dalam penelitian atau pengolahan data hanya akan datang kepada mereka yang menjaga integritas dan keadilan. Ayat ini juga mengajarkan bahwa keteguhan dalam memegang prinsip adalah kunci keberhasilan. Dalam dunia modern, di mana manipulasi data sering menjadi masalah serius, seorang muslim yang beriman harus berpegang teguh pada nilai islam untuk memastikan bahwa hasil yang diperoleh dapat dipertanggungjawabkan dan niat yang benar.

Allah SWT juga berfirman pada QS. An Nur ayat 24:

يَوْمَ تَشْهَدُ عَلَيْهِمُ السِّتَّةُمْ وَأَيْدِيهِمْ وَأَرْجُلُهُمْ بِمَا كَانُوا يَعْمَلُونَ ﴿٢٤﴾

(Kementerian Agama, 2022)

Artinya: “Pada hari ketika lidah, tangan, dan kaki mereka menjadi saksi atas apa yang dahulu mereka kerjakan”. (Q.S An Nur [24]: 24)

Ayat tersebut memberikan pelajaran mendalam tentang kejujuran dan tanggung jawab atas segala perbuatan manusia. Tidak hanya itu, dalam ayat ini Allah SWT menggambarkan hari kiamat, dimanamanusia tidak dapat berbohong atau berdalih karena mulut mereka akan ditutup, sementara tangan dan kaki akan menjadi saksi tentang segala perbuatan yang telah dilakukan. Ayat ini sangat relevan dengan proses klasifikasi biner dalam dunia data dan penelitian. Dalam klasifikasi biner, data bertindak sebagai “saksi” yang menentukan hasil

pengelompokkan menjadi dua kategori, seperti “benar” atau “salah”, “positif” atau “negatif”. Oleh karena itu, jika data yang digunakan jujur dan akurat, hasil yang diperoleh akan mencerminkan kebenaran. Sebaliknya, data yang salah atau dimanipulasi akan menghasilkan keputusan yang tidak tepat, yang berpotensi merugikan banyak pihak.

Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi biner menggunakan metode Gauss-Newton dan conjugate gradient pada regresi logistik. Dalam prosesnya, metode ini berfokus pada optimasi, yang memaksimalkan keakuratan klasifikasi berdasarkan data yang tersedia. Proses optimasi ini dapat di analogikan dengan konsep hisab sesuai dengan QS.An-Nur: 24, dimana setiap amal manusia dihisab berdasarkan fakta tanpa ada yang tersembunyi. Seperti halnya tangan dan kaki yang bersaksi jujur, data yang digunakan dalam penelitian harus mencerminkan realitas tanpa adanya manipulasi. Dengan menjaga integritas suatu data, penelitian ini menjadi bentuk tanggung jawab untuk mendapatkan hasil analisis yang bermanfaat dan terpercaya.

Ayat ini juga cukup relevan dengan kehidupan modern, terutama dalam bentuk konteks data dan teknologi. Dimana anggota tubuh akan menjadi saksi dan tidak bisa berkata dusta di akhirat dapat di analogikan dengan kejujuran suatu data dalam pengolahan saat ini. Memanipulasi data tidak hanya merugikan secara teknis saja, namun dapat menimbulkan dampak yang cukup besar. Dengan demikian, penelitian ini tidak hanya bertujuan untuk meningkatkan performa dari metode yang digunakan, akan tetapi juga mencerminkan nilai-nilai yang telah di ajarkan oleh Al-Quran yaitu nilai-nilai keadilan dan kejujuran.

## 2.6 Kajian Topik Dengan Teori Pendukung

Klasifikasi biner merupakan masalah penting dalam pembelajaran mesin dan statistik (Bishop, 2006). Regresi logistik adalah metode umum yang digunakan untuk menyelesaikan masalah klasifikasi biner, karena kemampuannya memodelkan probabilitas kejadian suatu kelas (Nocedal, J., & Wright, 2006). Namun, dalam proses optimasi parameter regresi logistik, metode ini sering digunakan untuk menentukan efisiensi dan akurasi hasil klasifikasi (Boyd, S., & Vandenberghe, 2004). Metode Gauss-Newton adalah pendekatan optimasi yang sering digunakan dalam regresi logistik, tetapi memiliki kelemahan dalam konvergensi yang bergantung pada kondisi awal dan tidak selalu optimal dalam menangani masalah dengan dimensi besar (Fletcher, 1987).

Di sisi lain, metode conjugate gradient dikenal lebih efisien dalam menangani sistem linear berskala besar tanpa perlu menyimpan matriks Hessian secara eksplisit (Nocedal, J., & Wright, 2006). Oleh karena itu, pendekatan gabungan antara Gauss-Newton dan conjugate gradient dapat meningkatkan performa optimasi dalam masalah klasifikasi biner menggunakan regresi logistik.

Regresi logistik dalam klasifikasi biner adalah model statistik yang digunakan untuk memprediksi probabilitas suatu sampel masuk dalam kelas tertentu berdasarkan sekumpulan fitur (Bishop, 2006). Fungsi logistik atau sigmoid digunakan untuk membatasi output dalam rentang (0,1) (Jamhuri et al., 2022).

$$h_i = \frac{1}{1 + \exp(-z_i)}$$

di mana:

$z_i$  adalah parameter model yang harus diestimasi dengan metode optimasi (Fletcher, 1987).

Metode Gauss-Newton dalam optimasi regresi logistik digunakan untuk menyelesaikan permasalahan estimasi parameter dalam model non-linear (Nocedal, J., & Wright, 2006). Dengan menggunakan pendekatan deret Taylor orde pertama pada fungsi residu, metode ini bekerja dengan iterasi sebagai berikut:

$$\Delta \vec{\theta} = [J^T J]^{-1} J^T D$$

di mana:

$A$  adalah matriks Jacobian dari residu  $B$  (Jamhuri et al., 2022). Salah satu kendala utama metode ini adalah kebutuhan untuk menghitung dan menyimpan matriks Hessian.

Metode conjugate gradient merupakan pendekatan iteratif yang digunakan untuk menyelesaikan sistem linear besar dengan bentuk  $Ax = b$  (Nocedal, J., & Wright, 2006). Dalam konteks Gauss-Newton, metode ini dapat digunakan untuk menyelesaikan sistem linear  $J^T J \Delta \theta = J^T D$  tanpa perlu menghitung invers secara eksplisit (Jamhuri et al., 2022). Dengan memanfaatkan metode conjugate gradient, iterasi Gauss-Newton dapat dipercepat dengan menyelesaikan langkah perbaikannya secara lebih efisien, terutama ketika diterapkan pada masalah dengan dimensi besar (Lan, 2004). Selain itu, metode ini memiliki keunggulan dalam hal penggunaan memori yang relatif rendah karena tidak perlu menyimpan matriks penuh, sehingga sangat sesuai untuk diterapkan pada permasalahan optimasi berskala besar dengan jumlah parameter yang banyak.

## BAB III METODE PENELITIAN

### 3.1 Jenis Penelitian

Jenis penelitian ini merupakan penelitian kuantitatif yang bertujuan untuk mengoptimalkan model klasifikasi biner menggunakan metode Gauss-Newton. Pendekatan optimasi yang diterapkan melibatkan dua metode iteratif, yaitu Gauss-Newton dan conjugate gradien, yang kemudian dibandingkan dengan metode optimasi lainnya seperti *Gradient Descent* dan Adam. Penelitian ini akan menguji kinerja metode-metode ini dalam menyelesaikan masalah klasifikasi biner pada dataset *Ionosphere*.

### 3.2 Data dan Sumber Data

Penelitian ini menggunakan dataset *Ionosphere* yang tersedia di *UCI Machine Learning Repository*. Dataset *Ionosphere* terdiri dari 351 sampel dan 34 fitur. Dataset ini terdiri dari ionosfer menggunakan sinyal yang diproses dengan fungsi autokorelasi. Data mengidentifikasi apakah ada struktur dalam sinyal yang dikembalikan oleh radar.

### 3.3 Struktur Data

Secara umum, struktur data untuk dataset yang digunakan dalam penelitian ini terdiri dari  $x$  dan  $y$  sebagai berikut

Tabel 3.1 Struktur Data Ionosfer

Fitur ( $X$ )	Label ( $Y$ )	Jumlah sample ( $m$ )
34	2	351

### 3.4 Tahapan Penelitian

Tahapan-tahapan penelitian yang akan dilakukan, antara lain:

#### 3.4.1 Persiapan Data

Sebelum dilakukan analisis, data harus disamakan fiturnya akan berada pada rentang 0 – 1

##### 1. Normalisasi fitur

Setelah pembersihan data dilakukan, langkah selanjutnya adalah normalisasi data. Langkah ini perlu dilakukan karena fitur-fitur pada dataset memiliki skala yang berbeda-beda. Untuk memastikan bahwa setiap fitur memiliki bobot yang setara, maka dilakukan normalisasi data menggunakan metode *min-max scalling*.

$$x'_{ij} = \frac{x_{ij} - \min(\vec{x}_j)}{\max(\vec{x}_j) - \min(\vec{x}_j)}$$

Di mana:

$\vec{x}_j$  adalah kolom ke- $j$  dari matriks fitur,

$x'_{ij}$  adalah nilai fitur setelah normalisasi.

$\min(\vec{x}_j)$  dan  $\max(\vec{x}_j)$  adalah masing-masing nilai minimum dan maksimum dari fitur  $j$  dalam dataset.

Dengan normalisasi, maka fitur akan berada pada rentang [0,1], sehingga kontribusi setiap fitur dalam perhitungan jarak menjadi setara.

##### 2. Pembagian dataset

Pada penelitian ini, dataset akan dibagi menjadi data latih (70%), validasi (15%), dan tes (15%). Pembagian ini dilakukan secara acak untuk memastikan distribusi kelas yang seimbang disetiap subset.

### 3.4.2 Pelatihan Model Regresi Logistik

Langkah-langkah dalam pelatihan model regresi logistik persamaan (2.1)

sebagai berikut:

1. Menghitung nilai  $z_i$ 
  - a. Ambil sampel data  $\vec{x}_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$ .
  - b. Kemudian inisialisasi bobot  $\vec{\theta} = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)$ .
  - c. Hitung  $z_i$  menggunakan persamaan

$$z_i = \vec{x}_i \cdot \vec{\theta} = \theta_0 + x_{i1}\theta_1 + x_{i2}\theta_2, \dots + x_{in}\theta_n$$

- b. Ulangi proses ini untuk seluruh sampel dalam dataset, sehingga diperoleh  $z_1, z_2, \dots, z_m$
2. Menghitung nilai  $h_i$

Setelah mendapat  $z_i$ , gunakan fungsi sigmoid untuk menghitung probabilitas  $h_i$ :

- a. Substitusi nilai  $z_i$  ke dalam fungsi sigmoid pada persamaan (2.1).
  - b. Ulangi langkah ini untuk semua  $z_i$  dari setiap sampel, sehingga diperoleh  $h_1, h_2, \dots, h_m$ .

3. Menghitung error untuk seluruh data

Setelah menghitung nilai  $h$ , maka langkah selanjutnya adalah menghitung error untuk seluruh data. Error dihitung berdasarkan selisih antara prediksi ( $h_i$ ) dan nilai aktual ( $y_i$ ) dengan langkah-langkah sebagai berikut:

- a. Gunakan fungsi error log los (*binary cross-entropy*) pada persamaan (2.3)
  - b. Hitung nilai  $L$  dengan menjumlahkan error dari seluruh sampel dalam dataset.

4. Meminimalkan fungsi error dengan update bobot

Setelah menghitung error, langkah selanjutnya adalah meminimalkan fungsi error menggunakan metode optimasi Gauss-Newton dan conjugate gradient.

### 3.4.3 Implementasi Metode Gauss-Newton

Untuk memperkirakan nilai parameter  $\vec{\theta}$  pada *loss function* pada persamaan (2.3).

1. Input nilai awal  $\vec{\theta}^{(0)}$  dengan sepasang data  $(\vec{x}_i, y_i)$ , Nilai Data  $m$ ; toleransi untuk menentukan konvergensi; nilai maksimum iterasi  $N$ .
2. Output Bobot optimal  $\vec{\theta}^{(N)}$ , atau pesan bahwa iterasi maksimum terlampaui.

Langkah-langkahnya sebagai berikut:

- a. Tetapkan  $k = 1$  dan gunakan bobot awal  $\theta_0$  sebagai awal parameter
- b. Selama  $k \leq N$  (belum melebihi iterasi maksimum), maka ulangi persamaan (5-8).
- c. Hitung matriks Jacobian  $A$  dan matriks residu  $B$  yang telah didefinisikan pada persamaan (18).
- d. Setelah menghitung  $A$  dan  $B$ , maka langkah selanjutnya adalah update bobot  $\Delta\vec{\theta} = [A^T A - A^T B \cdot [J^T J]^{-1} J^T D]$ . Namun, untuk menghitung invers matriks itu rumit, terutama untuk matriks yang berukuran besar. Untuk mengatasi hal ini, maka digunakan metode conjugate gradient, yang menyelesaikan perhitungan tanpa perlu menghitung invers secara langsung.

- e. Langkah selanjutnya adalah memperbarui bobot dengan menggunakan persamaan  $\vec{\theta}^{(k+1)} = \vec{\theta}^{(k)} + \Delta\vec{\theta}$ . Kemudian hitung selisih antara bobot baru dengan bobot lama  $\vec{\theta}^{(k+1)} - \vec{\theta}^{(k)}$ . Jika  $\vec{\theta}^{(k+1)} - \vec{\theta}^{(k)} \leq TOL$  (toleransi), maka model dianggap konvergen. dengan outputnya adalah  $\vec{\theta}^{(k+1)}$ . Jika dirasa sudah konvergen, maka hentikan iterasi ini.
- f. Jika belum konvergen, lanjutkan ke iterasi berikutnya dengan menggunakan  $k = k + 1$
- g. Jika  $k > N$  (jumlah iterasi melebihi batas maksimum) maka iterasi maksimum tercapai, solusi tidak ditemukan dan berhenti. Untuk meningkatkan performa pada metode Gauss-Newton, maka langkah selanjutnya yaitu menggunakan metode conjugate gradient yang dapat membantu mempercepat proses pencarian parameter optimal dengan mengurangi jumlah komputasi yang diperlukan, terutama saat menghadapi data yang berukuran besar.

#### 3.4.4 Implementasi Metode Conjugate Gradient

1. Inisialisasi
  - a. Tetapkan perkiraan awal  $\vec{\theta}_0$ .
  - b. Kemudian hitung residu awal untuk mengetahui seberapa jauh solusi awal  $\theta_0$  dari solusi yang diinginkan.

$$\vec{r}_0 = B - A\theta_0$$

- c. Setelah menghitung residu awal, selanjutnya adalah menetapkan arah awal. Gunakan residu awal sebagai arah pertama untuk iterasi.

2. Iterasi untuk  $k = 0, 1, 2, \dots$ , lakukan:
  - a. Kondisi berhenti  
Periksa apakah solusi sudah cukup akurat. Jika  $r_k = 0$  atau  $r_k <$  toleransi, maka iterasi dihentikan.
  - b. Hitung parameter langkah optimal. Hitung  $\alpha_k$  untuk menentukan seberapa jauh harus bergerak ke arah  $d_k$ . Menggunakan persamaan (2.2022).
  - c. Langkah selanjutnya adalah, memperbarui solusi dengan menggunakan  $\alpha_k$  untuk memperbarui solusi  $\theta_k$  sehingga lebih akurat ke solusi akhir sesuai dengan persamaan (2.2123).
  - d. hitung  $\vec{r}_{k+1}$  untuk mengetahui kesalahan setelah pembaruan solusi menggunakan persamaan (2.2224).
  - e. Kemudian hitung parameter conjugate dengan cara menghitung  $\beta_k$  untuk menyesuaikan arah langkah berikutnya menggunakan persamaan (2.2325).
  - f. Setelah menghitung  $\beta$ , Langkah selanjutnya adalah menghitung  $d$  menggunakan persamaan (2.2426).

### 3.4.5 Evaluasi Model

Setelah proses pengelompokkan data dilakukan dengan metode Gauss-Newton dan conjugate gradient, hasil pengelompokkan tersebut kemudian dievaluasi dan divalidasi melalui metrik evaluasi berbasis klasifikasi. Selanjutnya, hasil ini akan dibandingkan dengan metode dasar dan penelitian sebelumnya untuk melihat perbandingannya.

## 1. Metrik evaluasi

- a. Akurasi: Presentase prediksi yang benar dari total data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- b.
- Precision*
- : Rasio
- true positive*
- terhadap total prediksi positif

$$Precision = \frac{TP}{TP + FP}$$

- c.
- Recall*
- : Rasio
- true positive*
- terhadap total kasus sebenarnya yang positif.

$$Recall = \frac{TP}{TP + FN}$$

- d.
- F1-Score*
- : Harmonik antara
- precision*
- dan
- recall*
- agar keduanya saling seimbang.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Dimana:

TP (*True Positive*) : Kasus dimana model memprediksi positif dan kenyataannya juga positif,

TN (*True Negative*) : Kasus dimana model memprediksi negatif dan kenyataannya juga negatif.

FP (*False Positive*) : Kasus dimana model memprediksi positif namun kenyataannya positif.

FN (*False Negative*) : Kasus dimana model memprediksi negatif namun kenyataannya positif.

Kemudian selain itu, model akan dibandingkan dengan metode optimasi dasar, seperti:

- 1) *Gradient Descent*: Algoritma optimasi paling dasar, dengan pembaruan parameter berdasarkan turunan pertama dari fungsi biaya.
- 2) *Adam (Adaptive Moment Estimation)*: Algoritma optimasi yang menggunakan momen pertama dan kedua dari gradient untuk mempercepat konvergensi.

#### **3.4.6 Flowchart**

Proses pelatihan model klasifikasi biner pada penelitian ini disajikan secara sistematis melalui *flowchart* yang ditampilkan pada Gambar 3.1. *Flowchart* tersebut disusun berdasarkan arahan pembimbing serta referensi dari literatur yang telah dipelajari.



### **3.4.7 Perbandingan dengan Penelitian Sebelumnya**

Hasil optimasi dengan metode Gauss-Newton dan *conjugate gradient* akan dibandingkan dengan hasil-hasil dari penelitian sebelumnya.

## BAB IV

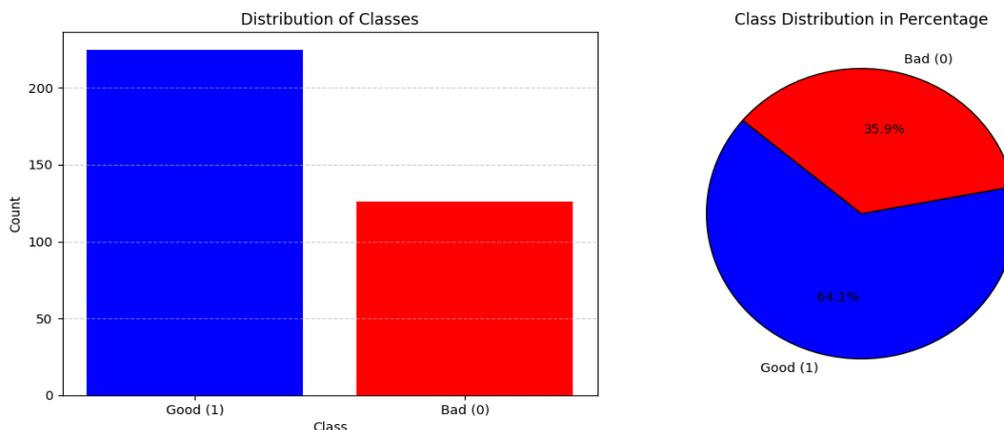
### HASIL DAN PEMBAHASAN

#### 4.1 Hasil

##### 4.1.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini adalah dataset *Ionosphere* yang diperoleh dari UCI Machine Learning Repository. Dataset ini terdiri dari 351 sampel dengan 34 fitur numerik yang merepresentasikan data sinyal radar, serta dilengkapi dengan label klasifikasi berupa kategori *bad* dan *good*. Pengolahan data dilakukan menggunakan bahasa pemrograman *Python* di lingkungan *Google Collaboratory* untuk memastikan proses yang efisien dan terstruktur. Seluruh data asli *Ionosphere* dapat dilihat secara lengkap pada lampiran 1.

Berikut ini adalah visualisasi perbandingan jumlah sampel berdasarkan kategori klasifikasi yang dapat dilihat pada Gambar 4.1. Dari gambar tersebut, terlihat bahwa pada dataset *Ionosphere*, kelas “*good*” mendominasi dengan proporsi sebesar 64,1%, sedangkan kelas “*bad*” memiliki proporsi sebesar 35,9% dari total seluruh data yang digunakan.



Gambar 4.1 Presentasi *Good* dan *Bad* pada Data *Ionosphere*

#### 4.1.2 Pra Pemrosesan Data

Pra-pemrosesan data merupakan tahap awal yang sangat penting dalam analisis data, dengan tujuan memastikan bahwa data sudah dalam kondisi yang sesuai untuk diterapkan ke dalam model pembelajaran mesin. Langkah pertama dalam pra-pemrosesan ini adalah mengonversi label kelas yang awalnya berupa huruf ‘g’ (*good*) dan ‘b’ (*bad*) menjadi format numerik, yaitu 1 untuk “*good*” dan 0 untuk “*bad*”. Selain itu, untuk meningkatkan keterbacaan dan memudahkan analisis, nama fitur pada dataset disajikan ulang secara sistematis dengan penamaan baru berupa  $x_1, x_2, \dots, x_n$ . Sebagai ilustrasi cara kerja pra-pemrosesan ini, ditampilkan 6 data dengan 2 fitur yang memiliki korelasi tinggi dari dataset *Ionosphere*, agar proses perhitungan manual dapat dilakukan.

Tabel 4.1 Cuplikan 6 Data yang Diambil dengan 2 Fitur dengan Korelasi Tinggi

$i$	$X$		$y$
	$x_2$	$x_4$	
1	0.9953	0.8524	1
2	1	0.9303	0
3	1	1	1
4	1	1	0
5	1	0.9414	1
6	0.0233	-0.0992	0

#### 4.1.3 Normalisasi Fitur

Proses normalisasi fitur dataset *Ionosphere* dilakukan untuk meningkatkan kualitas data sebelum analisis lebih lanjut. Langkah-langkah yang diterapkan dalam proses ini meliputi. Pertama, mengetahui nilai minimum dan maksimum dari masing-masing fitur dalam dataset. Hal ini dilakukan untuk setiap kolom fitur, baik untuk  $x_3$  atau  $x_5$ . Nilai minimum dan maksimum digunakan untuk

mengetahui rentang dari data yang ada dan merupakan dasar untuk melakukan normalisasi. Setelah memperoleh nilai minimum dan maksimum, langkah kedua yaitu menghitung nilai normalisasi untuk setiap data dengan rumus *Min-Max Scalling* dengan rumus berikut

$$x'_{ij} = \frac{x_{ij} - \min(\vec{x}_j)}{\max(\vec{x}_j) - \min(\vec{x}_j)}$$

Contoh perhitungan normalisasi fitur  $x_{ij}$  di ambil berdasarkan tabel 4.1 untuk  $i = 1, 2, \dots, 5$  dan  $j = 1$  dan 2 akan di definisikan sebagai berikut

$$\vec{x}_1 = (0.9953; 1; 1; 1; 1; 0.0233)$$

$$\vec{x}_2 = (0.8524; 0.9303; 1; 1; 0.9414; -0.0992)$$

$$\text{Min}(\vec{x}_1) = 0.0233$$

$$\text{Max}(\vec{x}_1) = 1$$

$$\text{Min}(\vec{x}_2) = -0.0992$$

$$\text{Max}(\vec{x}_2) = 1$$

1. Normalisasi untuk  $x_{i1}$

a. Untuk  $i = 1, j = 1$

$$\begin{aligned} x'_{11} &= \frac{x_{11} - \min(\vec{x}_1)}{\max(\vec{x}_1) - \min(\vec{x}_1)} \\ &= \frac{0.9953 - 0.0233}{1 - 0.0233} \\ &= 0.9953 \end{aligned}$$

b. Untuk  $i = 2, j = 1$

$$\begin{aligned} x'_{21} &= \frac{x_{21} - \min(\vec{x}_1)}{\max(\vec{x}_1) - \min(\vec{x}_1)} \\ &= \frac{1 - 0.0233}{1 - 0.0233} \end{aligned}$$

$$= 1$$

c. Untuk  $i = 3, j = 1$

$$\begin{aligned} x'_{31} &= \frac{x_{31} - \min(\vec{x}_1)}{\max(\vec{x}_1) - \min(\vec{x}_1)} \\ &= \frac{1 - 0.0233}{1 - 0.0233} \\ &= 1 \end{aligned}$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5, \text{ dan } 6$

2. Normalisasi untuk  $x_{i2}$

a. Untuk  $i = 1, j = 1$

$$\begin{aligned} x'_{12} &= \frac{x_{12} - \min(\vec{x}_2)}{\max(\vec{x}_2) - \min(\vec{x}_2)} \\ &= \frac{0.85243 - (-0.0992)}{1 - (-0.0992)} \\ &= 0.8658 \end{aligned}$$

b. Untuk  $i = 2, j = 1$

$$\begin{aligned} x'_{22} &= \frac{x_{22} - \min(\vec{x}_2)}{\max(x_2) - \min(\vec{x}_2)} \\ &= \frac{0.93035 - (-0.0992)}{1 - (-0.0992)} \\ &= 1 \end{aligned}$$

c. Untuk  $i = 3, j = 1$

$$\begin{aligned} x'_{32} &= \frac{x_{32} - \min(\vec{x}_2)}{\max(x_2) - \min(\vec{x}_2)} \\ &= \frac{1 - (-0.0992)}{1 - (-0.0992)} \\ &= 1 \end{aligned}$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5$ , dan 6

### 3. Hasil Normalisasi

Setelah proses normalisasi data dilakukan, nilai hasil normalisasi dari tabel 4.1 dapat disajikan dalam tabel berikut. Sedangkan hasil normalisasi dari seluruh data Ionosphere, yang digunakan dalam proses pelatihan model, dapat ditemukan pada lampiran 2.

Tabel 4.2 Tabel Normalisasi 6 Data dengan 2 Fitur

$i$	$X$		$y$
	$x_2$	$x_4$	
1	0.9953	0.8658	1
2	1	0.9366	0
3	1	1	1
4	1	1	0
5	1	0.9467	1
6	0	0	0

Nilai-nilai pada data tabel 4.2 akan digunakan sebagai data input untuk proses pelatihan menggunakan algoritma regresi logistik yang akan dijelaskan pada sub-bab berikutnya.

## 4.2 Inisialisasi Bobot

Setelah proses normalisasi fitur dilakukan untuk menyetarakan skala data, langkah berikutnya adalah melakukan inisialisasi bobot. Inisialisasi bobot merupakan tahap awal dalam proses pelatihan model, di mana setiap bobot diberikan nilai awal sebelum dilakukan proses iterasi untuk menemukan nilai bobot optimal. Nilai bobot awal ini dapat dipilih secara acak atau sesuai kebutuhan metodenya yaitu,  $\vec{\theta} = 0.1, 0.2, 0.3$ .

Setelah proses inialisasi bobot dilakukan, langkah selanjutnya yaitu menghitung nilai  $z_i$  untuk setiap data. Nilai  $z_i$  diperoleh dari hasil perkalian antara nilai fitur dengan bobot masing-masing, ditambah dengan nilai bias ( $x_0$ ). Namun, dalam tabel 4.1, nilai  $x_0$  sebagai bias tidak dicantumkan secara eksplisit. Hal ini dikarenakan nilai bias bersifat konstan untuk setiap data dan tidak termasuk dalam variabel input yang berasal dari dataset. Bias biasanya disimbolkan sebagai  $x_0 = 1$  dalam perhitungan, sehingga hanya terlibat dalam proses komputasi tanpa perlu dicantumkan dalam tabel data fitur tanpa perlu dicantumkan dalam tabel data fitur. Meski demikian, dalam proses perhitungan nilai  $z_i$ , bias tetap diikutsertakan dengan cara dikalikan bobot bias ( $\theta_0$ ) dan dijumlahkan ke hasil perkalian antara fitur dan bobot lainnya. Dengan demikian, perhitungan tetap berjalan lengkap, hanya saja nilai  $x_0$  tidak perlu ditampilkan dalam tabel karena bernilai konstan dan tidak memengaruhi variasi data fitur yang ditampilkan.

1. Menghitung  $z_i$

1. Menghitung  $z_i$  untuk  $i = 1, 2, \dots, 6$  menggunakan persamaan berikut

$$z_i = \vec{x}_i \cdot \vec{\theta}$$

- a. untuk  $i = 1$ ,

$$\begin{aligned} z_1 &= \theta_0 + x_{11} \times \theta_1 + x_{12} \times \theta_2 \\ &= 0.1 + 0.9952 \times 0.2 + 0.8657 \times 0.3 \\ &= 0.5588 \end{aligned}$$

- b. untuk  $i = 2$

$$\begin{aligned} z_2 &= \theta_0 + x_{21} \times \theta_1 + x_{22} \times \theta_2 \\ &= 0.1 + 1 \times 0.2 + 0.528 \times 0.3 \\ &= 0.581 \end{aligned}$$

c. untuk  $i = 3$

$$\begin{aligned} z_3 &= \theta_0 + x_{31} \times \theta_1 + x_{32} \times \theta_2 \\ &= 0.1 + 1 \times 0.2 + 0.9366 \times 0.3 \\ &= 0.6 \end{aligned}$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5$ , dan  $6$

2. Setelah seluruh nilai  $z_i$  untuk  $i = 1, 2, \dots, 6$  diperoleh, Langkah selanjutnya adalah menghitung  $h_i$  untuk masing-masing data menggunakan persamaan

$$h_i = \frac{1}{1 + \exp(-z_i)}$$

a. untuk  $i = 1$

$$\begin{aligned} h_1 &= \frac{1}{1 + \exp(-z_1)} \\ &= \frac{1}{1 + \exp(-0.5587)} \\ &= 0.6362 \end{aligned}$$

b. untuk  $i = 2$

$$\begin{aligned} h_2 &= \frac{1}{1 + \exp(-z_2)} \\ &= \frac{1}{1 + \exp(-0.5809)} \\ &= 0.6413 \end{aligned}$$

c. untuk  $i = 3$

$$h_3 = \frac{1}{1 + \exp(-z_3)}$$

$$= \frac{1}{1 + \exp(-0.6)}$$

$$= 0.6457$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5$ , dan  $6$

### 4.3 Hitung fungsi Loss

Setelah mendapatkan nilai  $h_i$  berikutnya menghitung error untuk setiap sample data, menggunakan fungsi *loss* berikut

$$L_i = -((y_i \log h_i) + (1 - y_i) \log (1 - h_i))$$

Langkah selanjutnya, yaitu menghitung  $L_i$  dengan  $i = 1, 2, \dots, 6$  menggunakan persamaan

1. Untuk  $i = 1$

$$L_1 = -((y_1 \times \log (h_1)) + (1 - y_1) \times \log (1 - h_1))$$

$$= -((1 \times \log(0.6361)) + (1 - 1) \times \log(1 - 0.6362))$$

$$= 0.4523$$

2. Untuk  $i = 2$

$$L_2 = -(y_2 \times \log(h_2) + (1 - y_2) \times \log(1 - h_2))$$

$$= -((0 \log (0.6412)) + (1 - 0) \log (1 - 0.6412))$$

$$= 1.0253$$

3. Untuk  $i = 3$

$$L_3 = -(y_3 \times \log (h_3) + (1 - y_3) \times \log (1 - h_3))$$

$$= -((1 \log (0.6457)) + (1 - 1) \log (1 - 0.6457))$$

$$= 0.4375$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5$ , dan  $6$

Setelah menghitung nilai loss untuk masing-masing data sampel, langkah berikutnya adalah menghitung nilai gradien untuk setiap parameter bobot yang akan di bahas pada tahap selanjutnya.

#### 4.4 Gauss Newton

Perhitungan  $\nabla L_i$  dilakukan dengan mengambil turunan dari fungsi loss terhadap masing-masing bobot, sehingga diperoleh nilai gradien untuk setiap fitur pada data ke  $i$ . Nilai gradien inilah yang akan digunakan dalam proses pembaruan bobot pada tahap optimasi berikutnya. Selanjutnya, nilai gradien  $\nabla L_i$  untuk setiap data dihitung berdasarkan turunan fungsi loss terhadap bobot, dengan persamaan sebagai berikut

$$\frac{\partial L(\vec{\theta})}{\partial \theta_j} = (y_i - h_i)x_{ij}$$

1. Menghitung  $\nabla L_1$  untuk  $x_{i1}$

a. Untuk  $i = 1, j = 1$

$$\begin{aligned} \frac{\partial L(\vec{\theta})}{\partial \theta_1} &= (y_1 - h_1) x_{11} \\ &= (1 - 0.6361) \times 1 \\ &= 0.3638 \end{aligned}$$

b. Untuk  $i = 2, j = 1$

$$\begin{aligned} \frac{\partial L(\vec{\theta})}{\partial \theta_1} &= (y_2 - h_2) x_{21} \\ &= (0 - 0.6412) \times 1 \\ &= -0.6412 \end{aligned}$$

c. Untuk  $i = 3, j = 1$

$$\begin{aligned}\frac{\partial L(\vec{\theta})}{\partial \theta_1} &= (y_3 - h_3) x_{31} \\ &= (1 - 0.6456) \times 1 \\ &= 0.3543\end{aligned}$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5$ , dan 6

2. Menghitung  $\nabla L_2$  untuk  $x_{i2}$

a. Untuk  $i = 1, j = 2$

$$\begin{aligned}\frac{\partial L(\vec{\theta})}{\partial \theta_2} &= (y_1 - h_1) x_{12} \\ &= (1 - 0.6361) \times 0.9953 \\ &= 0.3621\end{aligned}$$

b. Untuk  $i = 2, j = 2$

$$\begin{aligned}\frac{\partial L(\vec{\theta})}{\partial \theta_2} &= (y_2 - h_2) x_{22} \\ &= (1 - 0.6361) \times 1 \\ &= -0.6413\end{aligned}$$

c. Untuk  $i = 3, j = 2$

$$\begin{aligned}\frac{\partial L(\vec{\theta})}{\partial \theta_2} &= (y_3 - h_3) x_{32} \\ &= (1 - 0.6456) \times 1 \\ &= 0.3543\end{aligned}$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5$ , dan 6

3. Menghitung  $\nabla L_3$  untuk  $x_{i3}$

a. Untuk  $i = 1, j = 3$

$$\begin{aligned}\frac{\partial L(\vec{\theta})}{\partial \theta_3} &= (y_1 - h_1) x_{13} \\ &= (1 - 0.6361) \times 0.8524 \\ &= 0.315\end{aligned}$$

b. Untuk  $i = 2, j = 3$

$$\begin{aligned}\frac{\partial L(\vec{\theta})}{\partial \theta_3} &= (y_2 - h_2) x_{23} \\ &= (1 - 0.6361) \times 0.9303 \\ &= -0.6006\end{aligned}$$

c. Untuk  $i = 3, j = 3$

$$\begin{aligned}\frac{\partial L(\vec{\theta})}{\partial \theta_3} &= (y_3 - h_3) x_{33} \\ &= (1 - 0.6456) \times 1 \\ &= 0.3543\end{aligned}$$

Dengan cara yang sama dilakukan untuk  $i = 4, 5$ , dan 6

#### 4.5 Membangun Matriks Jacobian dan Matriks RHS (*Right Hand Side*)

Setelah menghitung nilai gradien untuk setiap parameter bobot, langkah berikutnya dalam proses optimasi model regresi logistik adalah membangun Matriks Jacobian dan Matriks RHS (*Right Hand Side*). Kedua matriks ini merupakan komponen penting dalam metode optimasi seperti Gauss-Newton dan conjugate gradient, yang digunakan untuk memperbarui bobot hingga mencapai solusi optimal.

Dengan demikian, matriks jacobian mempresentasikan seberapa besar perubahan fungsi error terhadap masing-masing parameter model. Ketika dilakukan perhitungan menggunakan data yang tersedia, diperoleh hasil matriks Jacobian sebagai berikut

$$J = \begin{bmatrix} 0.3638 & 0.3621 & 0.315 \\ -0.6413 & -0.6413 & -0.6006 \\ 0.3543 & 0.3543 & 0.3543 \\ -0.6457 & -0.6457 & -0.6457 \\ 0.3580 & 0.3580 & 0.3389 \\ -0.525 & 0 & 0 \end{bmatrix}$$

Setelah matriks jacobian terbentuk, langkah selanjutnya yaitu memebentuk matriks RHS, matriks RHS ini terebntuk oleh hasil dari perhitungan fungsi *loss* sebagai berikut

$$D = \begin{bmatrix} 0.4523 \\ 1.0252 \\ 0.4374 \\ 1.0375 \\ 0.4432 \\ 0.7444 \end{bmatrix}$$

Setelah membentuk matriks jacobian dan membentuk matriks RHS D, maka langkah selanjutnya adalah melakukan transpose terhadap matriks jacobian untuk menghitung nilai  $J^T J$  dan  $J^T D$  yang akan digunakan dalam sistem persamaan linier tersebut

$$J^T = \begin{bmatrix} 0.3638 & -0.6412 & 0.3543 & -0.6456 & 0.3580 & -0.5249 \\ 0.3629 & -0.6412 & 0.3543 & -0.6456 & 0.3580 & 0 \\ 0.3149 & -0.6006 & 0.3543 & -0.6456 & 0.3389 & 0 \end{bmatrix}$$

Setelah mendapatkan matriks jacobian dan matriks RHS D, langkah selanjutnya yaitu menghitung matriks *Hessian Aproximation* dengan persamaan berikut

$$J^T J$$

Hasil perhitungan  $J^T J$  adalah sebagai berikut

$$J^T J = \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix}$$

Setelah mendapatkan nilai  $J^T J$ , maka langkah selanjutnya yaitu menghitung nilai  $J^T D$  dengan menggunakan persamaan berikut

$$J^T D$$

Maka diperoleh

$$B = \begin{bmatrix} 0.3638 & -0.6412 & 0.3543 & -0.6456 & 0.3580 & -0.5249 \\ 0.3629 & -0.6412 & 0.3543 & -0.6456 & 0.3580 & 0 \\ 0.3149 & -0.6006 & 0.3543 & -0.6456 & 0.3389 & 0 \end{bmatrix} \begin{bmatrix} 0.4522 \\ 1.0252 \\ 0.4374 \\ 1.0374 \\ 0.4431 \\ 0.7443 \end{bmatrix}$$

$$= \begin{bmatrix} -1.24 \\ -0.8499 \\ -0.838 \end{bmatrix}$$

Setelah memperoleh nilai  $J^T J$  dan  $J^T D$ , langkah selanjutnya adalah membentuk sistem persamaan linier yang digunakan dalam pendekatan metode Gauss-newton. pada tahap ini ditetapkan bahwa  $\theta = \Delta\theta$ ,  $A = J^T J$ , dan  $B = J^T D$ . Sehingga, sistem persamaan linier yang diperoleh dapat dituliskan sebagai

$$A\theta = B$$

maka sistem persamaan linier yang terbentuk dapat diselesaikan menggunakan metode conjugate gradient. Metode ini digunakan untuk menghitung  $\theta$  secara iteratif tanpa perlu melakukan invers matriks secara langsung.

#### 4.6 Penyelesaian Sistem dengan Metode Conjugate Gradient

Setelah mendapatkan matriks *Hessian Approximation*  $A$  dan  $B$ , maka langkah selanjutnya adalah memasuki proses iterasi conjugate gradient. Proses iterasi ini

bertujuan untuk mencari nilai  $\theta$  yang memenuhi persamaan  $A\theta = B$  dengan persamaan berikut

1. Inisialisasi

Ditetapkan tebakan awal parameter

$$\theta_0 = \begin{bmatrix} 0.01 \\ 0.01 \\ 0.01 \end{bmatrix}$$

Setelah menetapkan  $\theta_0$ , maka langkah selanjutnya adalah menghitung  $r_k$ , untuk  $k = 0$ , dan  $\varepsilon = 0.0001$

2. Hitung  $r_0$

$$\begin{aligned} r_0 &= B - A\theta_0 \\ &= \begin{bmatrix} -1.24 \\ -0.8499 \\ -0.838 \end{bmatrix} - \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix} \begin{bmatrix} 0.01 \\ 0.01 \\ 0.01 \end{bmatrix} \\ &= \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix} \end{aligned}$$

3. Setelah menghitung  $r_0$ , langkah selanjutnya adalah menetapkan  $d_0$ . Dengan menggunakan persamaan

$$d_0 = r_0$$

Sehingga diperoleh

$$d_0 = \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix}$$

Sehingga  $r_0^T = [-1.2787 \quad -0.8857 \quad -0.8724]$

4. Setelah menetapkan  $d_0$ , langkah selanjutnya yaitu menghitung  $\alpha_k$  dengan menggunakan persamaan (2.22).

$$\alpha_0 = \frac{\vec{r}_0^T \vec{r}_0}{\vec{d}_0^T A \vec{d}_0}$$

$$= \frac{\begin{bmatrix} -1.2787 & -0.8857 & -0.8724 \end{bmatrix} \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix}}{\begin{bmatrix} -1.2787 & -0.8857 & -0.8724 \end{bmatrix} \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix} \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix}}$$

$$= 0.2794$$

5. Setelah menghitung  $\alpha_0$ , langkah selanjutnya adalah menghitung  $\theta_1$  berdasarkan persamaan (2.23)

$$\begin{aligned} \theta_1 &= \theta_0 + \alpha_0 \vec{d}_0 \\ &= \begin{bmatrix} 0.01 \\ 0.01 \\ 0.01 \end{bmatrix} + 0.2798 \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix} \\ &= \begin{bmatrix} -0.3473 \\ -0.2375 \\ -0.2338 \end{bmatrix} \end{aligned}$$

6. Setelah menghitung  $\theta_1$ , Langkah selanjutnya yaitu menghitung  $\vec{r}_1$  baru dengan menggunakan persamaan (2.24).

$$\begin{aligned} \vec{r}_1 &= \vec{r}_0 - \alpha_0 A \vec{d}_0 \\ &= \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix} - 0.2798 \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix} \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix} \\ &= \begin{bmatrix} -0.1620 \\ 0.1317 \\ 0.1037 \end{bmatrix} \end{aligned}$$

$$\text{Sehingga } r_1^T = [-0.1620 \quad 0.1317 \quad 0.1037]$$

7. Setelah menghitung  $\vec{r}_1$ , Langkah selanjutnya yaitu menghitung  $\|\vec{r}_1\|$  vektor dari residu

$$\begin{aligned} \|r_0\| &= \sqrt{x_1^2 + x_2^2 + x_3^2} \\ &= \sqrt{(-0.1620)^2 + (0.1317)^2 + (0.1037)^2} \end{aligned}$$

$$= \sqrt{0.0545}$$

$$= 0.2332$$

Iterasi akan dihentikan apabila  $\|r_1\| < 0.001$ . Jika  $\|r_0\| > 0.001$ , maka iterasi dilanjutkan dengan menghitung  $\beta, d$  sampai konvergen.

8. Setelah menghitung  $\|r_1\|$ , Langkah selanjutnya menghitung  $\beta$  dengan menggunakan persamaan (2.25).

$$\beta_0 = \frac{r_1^T r_1}{r_0^T r_0}$$

$$= \frac{\begin{bmatrix} -0.1620 & 0.1317 & 0.1037 \end{bmatrix} \begin{bmatrix} -0.1620 \\ 0.1317 \\ 0.1037 \end{bmatrix}}{\begin{bmatrix} -1.2787 & -0.8857 & -0.8724 \end{bmatrix} \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix}}$$

$$= 0.0171$$

9. Setelah mendapatkan nilai dari  $\beta_0$ , maka langkah selanjutnya adalah menghitung  $d_1$  dengan menggunakan persamaan (2.26)

$$d_1 = r_1 + \beta_0 d_0$$

$$= \begin{bmatrix} -0.1620 \\ 0.1317 \\ 0.1037 \end{bmatrix} + 0.0171 \begin{bmatrix} -1.2787 \\ -0.8858 \\ -0.8724 \end{bmatrix}$$

$$= \begin{bmatrix} -0.1839 \\ 0.1166 \\ 0.0888 \end{bmatrix}$$

Sehingga  $d_1^T = [-0.1839 \quad 0.1166 \quad 0.0888]$

Iterasi 2

Untuk  $k = 1$

1. Hitung  $\alpha$

$$\begin{aligned}\alpha_1 &= \frac{\vec{r}_1^T \vec{r}_1}{\vec{d}_1^T A \vec{d}_1} \\ &= \frac{\begin{bmatrix} -0.1620 & 0.1317 & 0.1037 \end{bmatrix} \begin{bmatrix} -0.1620 \\ 0.1317 \\ 0.1037 \end{bmatrix}}{\begin{bmatrix} -0.1839 & 0.1166 & 0.0888 \end{bmatrix} \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix} \begin{bmatrix} -0.1839 \\ 0.1166 \\ 0.0888 \end{bmatrix}} \\ &= 5.5901\end{aligned}$$

2. Hitung  $\Theta$

$$\begin{aligned}\Theta_2 &= \theta_1 + \alpha_1 \vec{d}_1 \\ &= \begin{bmatrix} 0.01 \\ 0.01 \\ 0.01 \end{bmatrix} + 5.5901 \begin{bmatrix} -0.1839 \\ 0.1166 \\ 0.0888 \end{bmatrix} \\ &= \begin{bmatrix} -1.3756 \\ 0.4144 \\ 0.2625 \end{bmatrix}\end{aligned}$$

3. Hitung  $r_2$

$$\begin{aligned}\vec{r}_2 &= \vec{r}_1 - \alpha_1 A \vec{d}_1 \\ &= \begin{bmatrix} -0.1620 \\ 0.1317 \\ 0.1037 \end{bmatrix} - 5.5901 \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix} \begin{bmatrix} -0.1839 \\ 0.1166 \\ 0.0888 \end{bmatrix} \\ &= \begin{bmatrix} 0.0009 \\ 0.0113 \\ -0.0129 \end{bmatrix}\end{aligned}$$

Sehingga  $r_2^T = [0.0009 \quad 0.0113 \quad -0.0129]$

4. Hitung  $\|r_2\|$

$$\begin{aligned}\|r_2\| &= \sqrt{x_1^2 + x_2^2 + x_3^2} \\ &= \sqrt{(0.0009)^2 + (0.0113)^2 + (-0.0129)^2} \\ &= \sqrt{0.002895} \\ &= 0.0172\end{aligned}$$

$\|r_2\| > 0.0001$ , maka iterasi dilanjutkan dengan menghitung  $\alpha, \theta, r, \beta, d$

5. Hitung  $\beta$

$$\begin{aligned}\beta_1 &= \frac{r_2^T r_2}{r_1^T r_1} \\ &= \frac{\begin{bmatrix} 0.0009 & 0.0113 & -0.0129 \end{bmatrix} \begin{bmatrix} 0.0009 \\ 0.0113 \\ -0.0129 \end{bmatrix}}{\begin{bmatrix} -0.1620 & 0.1317 & 0.1037 \end{bmatrix} \begin{bmatrix} -0.1620 \\ 0.1317 \\ 0.1037 \end{bmatrix}} \\ &= 0.0054\end{aligned}$$

6. Hitung  $d_2$

$$\begin{aligned}d_2 &= r_2 + \beta_1 d_1 \\ &= \begin{bmatrix} 0.0009 \\ 0.0113 \\ -0.0129 \end{bmatrix} + 0.0054 \begin{bmatrix} -0.1839 \\ 0.1166 \\ 0.0888 \end{bmatrix} \\ &= \begin{bmatrix} -3.7000e-05 \\ 1.2004e-02 \\ -1.2441e-02 \end{bmatrix}\end{aligned}$$

Sehingga  $d_2^T = [-3.7000e-05 \quad 1.2004e-02 \quad -1.2441e-02]$

Iterasi 3

Untuk  $k = 2$

1. Hitung  $\alpha$

$$\alpha_2 = \frac{\vec{r}_2^T \vec{r}_2}{\vec{d}_2^T A \vec{d}_2}$$

$$= \frac{\begin{bmatrix} 0.0009 & 0.0113 & -0.0129 \end{bmatrix} \begin{bmatrix} 0.0009 \\ 0.0113 \\ -0.0129 \end{bmatrix}}{\begin{bmatrix} -3.7000e-05 & 1.2004e-02 & -1.2441e-02 \end{bmatrix} \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix} \begin{bmatrix} -3.7000e-05 \\ 1.2004e-02 \\ -1.2441e-02 \end{bmatrix}}$$

$$= 852.4293$$

2. Hitung  $\Theta$

$$\Theta_3 = \theta_2 + \alpha_2 \vec{d}_2$$

$$= \begin{bmatrix} 0.01 \\ 0.01 \\ 0.01 \end{bmatrix} + 852.4293 \begin{bmatrix} -3.7000e-05 \\ 1.2004e-02 \\ -1.2441e-02 \end{bmatrix}$$

$$= \begin{bmatrix} -1.4071 \\ 10.6470 \\ -10.3424 \end{bmatrix}$$

3. Hitung  $r$

$$\vec{r}_3 = \vec{r}_2 - \alpha_2 A \vec{d}_2$$

$$= \begin{bmatrix} 0.0009 \\ 0.0113 \\ -0.0129 \end{bmatrix} - 852.4293 \begin{bmatrix} 1.4899 & 1.2136 & 1.1635 \\ 1.2136 & 1.213 & 1.163 \\ 1.1635 & 1.163 & 1.1173 \end{bmatrix} \begin{bmatrix} -3.7000e-05 \\ 1.2004e-02 \\ -1.2441e-02 \end{bmatrix}$$

$$= \begin{bmatrix} -0.0312 \\ -0.0287 \\ -0.0276 \end{bmatrix}$$

$$\text{Sehingga } r_3^T = [-0.0312 \quad -0.0287 \quad -0.0276]$$

4. Hitung  $\beta$

$$\beta_2 = \frac{r_3^T r_3}{r_2^T r_2}$$

$$\begin{aligned}
&= \frac{\begin{bmatrix} -0.0312 & -0.0287 & -0.0276 \end{bmatrix} \begin{bmatrix} -0.0312 \\ -0.0287 \\ -0.0276 \end{bmatrix}}{\begin{bmatrix} 0.0009 & 0.0113 & -0.0129 \end{bmatrix} \begin{bmatrix} 0.0009 \\ 0.0113 \\ -0.0129 \end{bmatrix}} \\
&= 8.6445
\end{aligned}$$

5. Hitung  $d$

$$\begin{aligned}
d_3 &= r_3 + \beta_2 d_2 \\
&= \begin{bmatrix} -0.0312 \\ -0.0287 \\ -0.0276 \end{bmatrix} + 8.6445 \begin{bmatrix} -3.7000e - 05 \\ 1.2004e - 02 \\ -1.2441e - 02 \end{bmatrix} \\
&= \begin{bmatrix} -0.0315 \\ 0.0750 \\ -0.1351 \end{bmatrix}
\end{aligned}$$

Dengan menggunakan langkah-langkah yang sama, dilakukan sampai dengan konvergen atau  $\|r\| < \varepsilon$ . Dengan menggunakan bantuan *python* 6 data dengan 2 fitur diperoleh bobot optimal  $[-1.4140 \quad 10.6688 \quad -10..3764]$ . Setelah bobot optimal diperoleh, langkah selanjutnya yaitu menghitung jumlah *loss* dari 6 sebagai berikut

$$\begin{aligned}
L(\vec{\theta}) &= \frac{1}{m} \sum_{i=1}^m L_i \\
&= \frac{1}{6} (0.4523 + 1.0253 + 0.4375 + 1.0375 + 0.4432 + 0.7444) \\
&= \frac{1}{6} \times 4.1402 \\
&= 0.69
\end{aligned}$$

Dengan menggunakan  $\delta = 0.1$ , jika  $|L| < \delta$  maka iterasi selesai. Jika tidak maka lanjut ke *epoch* selanjutnya. Dengan cara yang sama dilakukan sampai  $|L| < \delta$ . Setelah itu dapat mealnjutkan ke tahap berikutnya, yaitu dengan menggunakan

kurva pembelajaran yang memvisualisasikan perubahan nilai *loss* dan akurasi terhadap jumlah epoch. Kurva pembelajaran sangat berguna untuk mengamati apakah model mengalami *overfitting*, *underfitting*, atau telah mencapai konvergensi dengan baik.

Perlu ditegaskan bahwa proses evaluasi model ini tidak lagi terbatas pada 6 data dengan 2 fitur saja. Melainkan menggunakan keseluruhan data Ionosphere. Langkah ini bertujuan untuk mengevaluasi performa model secara menyeluruh terhadap seluruh variasi data. Sehingga hasil evaluasi dapat mempresentasikan kemampuan generalisasi model dalam klasifikasi data.

#### 4.7 Analisis Kurva Pembelajaran

Tabel berikut menunjukkan hasil akurasi dan loss model ionosphere pada data training dan validasi selama 100 epoch

Tabel 4.3 Perbandingan Nilai *Training Loss* pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih *Loss* Antara GN, Adam, dan GD

<i>Epoch</i>	<i>GN</i>	<i>GD</i>	<i>ADAM</i>	<i>Selisih GN – Adam</i>	<i>Selisih GN – GD</i>
1	0.7236	0.7417	0.7397	-0.0161	-0.0181
20	0.5566	0.6463	0.6514	-0.0948	-0.0897
40	0.4365	0.6194	0.6087	-0.1722	-0.1829
60	0.3912	0.6073	0.5781	-0.1869	-0.2161
80	0.3722	0.5976	0.5515	-0.1793	-0.2254
100	0.3606	0.5886	0.7397	-0.3791	-0.2280

Tabel 4.4 Perbandingan Nilai *Training Accuracy* pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih *Accuracy* Antara GN, Adam, dan GD

<i>Epoch</i>	<i>GN</i>	<i>GD</i>	<i>ADAM</i>	<i>Selisih GN – Adam</i>	<i>Selisih GN – GD</i>
1	0.4040	0.4000	0.4000	0.0040	0.0040
20	0.7976	0.6129	0.6075	0.1901	0.1847
40	0.9155	0.6445	0.6446	0.2709	0.2710
60	0.9364	0.5825	0.6707	0.2657	0.3539
80	0.9382	0.6546	0.6995	0.2387	0.2836
100	0.9382	0.6611	0.7293	0.2089	0.2771

Tabel 4.5 Perbandingan Nilai *Validation Loss* pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih *Loss* Antara GN, Adam, dan GD

<i>Epoch</i>	<i>GN</i>	<i>GD</i>	<i>ADAM</i>	<i>Selisih GN – Adam</i>	<i>Selisih GN – GD</i>
1	0.7121	0.7312	0.7294	-0.0173	-0.0191
20	0.5459	0.6262	0.6287	-0.0828	-0.0803
40	0.4287	0.5956	0.5845	-0.1558	-0.1669
60	0.3824	0.5825	0.5514	-0.1690	-0.2001
80	0.3637	0.5720	0.5225	-0.1588	-0.2083
100	0.3532	0.5625	0.4975	-0.1443	-0.2093

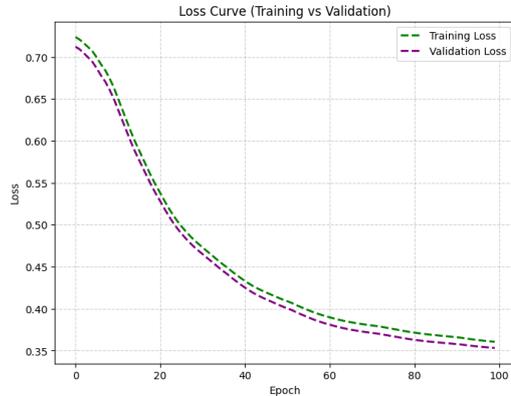
Tabel 4.6 Perbandingan Nilai *Validation Accuracy* pada Setiap Epoch untuk Metode GN, GD, dan Adam, Beserta Selisih *Loss* Antara GN, Adam, dan GD

<i>Epoch</i>	<i>GN</i>	<i>GD</i>	<i>ADAM</i>	<i>Selisih GN – Adam</i>	<i>Selisih GN – GD</i>
1	0.4528	0.4100	0.4150	0.0378	0.0428
20	0.9576	0.6124	0.6093	0.3483	0.3452
40	0.9247	0.6379	0.6447	0.28	0.2868
60	0.9576	0.6410	0.6690	0.2886	0.3166
80	0.9694	0.6537	0.7034	0.2660	0.3157
100	0.9778	0.6595	0.7342	0.2436	0.3183

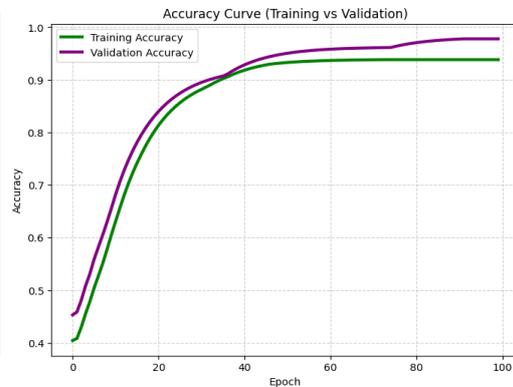
Berdasarkan Tabel 4.3, model mengalami peningkatan secara bertahap dengan penurunan *loss* yang relatif stabil. Nilai *loss* ini di dapatkan dari persamaan (2.3) yang umum digunakan dalam masalah klasifikasi biner. Berdasarkan tabel 4.4, hal ini menunjukkan bahwa proses pelatihan berjalan dengan baik, dimana model mampu mempelajari pola data tanpa mengalami *overfitting*, terlihat dari nilai *loss* pada data training dan validasi yang menurun secara seimbang.

Perlu dipahami bahwa tabel 4.4 merupakan hasil *training accuracy* selama proses pelatihan yang diperoleh secara otomatis pada setiap epoch, berdasarkan hasil prediksi model terhadap seluruh data. Nilai ini dihitung oleh *Python* pada setiap siklus epoch sebagai bagian dari kurva pembelajaran. Berdasarkan tabel 4.4 dapat dilihat bahwa model mencapai akurasi tertinggi sebesar 0.9382 pada data training tabel 4.6 dan 0.9778 pada data validasi pada epoch ke-100. Pencapaian ini menunjukkan bahwa model memiliki performa yang baik dalam melakukan

klasifikasi data *Ionosphere*. Adapun grafik hasil loss training dan validasi sebagai ditampilkan pada gambar berikut



Gambar 4.3 Perbandingan Nilai *Loss* pada Data Training dan Validasi Selama 100 epoch



Gambar 4.2 Perbandingan Nilai *Accuracy* pada Data Training dan Validasi Selama 100 Epoch

#### 4.8 Hasil Evaluasi Model pada Data Uji

Berikut merupakan hasil *confusion matrix* yang diperoleh dari model klasifikasi dengan data testing menggunakan algoritma Gauss-Newton. Ditunjukkan pada Tabel 4.4 berikut ini.

Tabel 4.7 Tabel Confusion Matriks

Confusion matrix	Prediksi	
	Positif	Negatif
Positif	34	0
Negatif	7	12

Tabel *confusion matrix* yang diberikan menunjukkan hubungan antara hasil prediksi model dan kondisi actual data. Baris pada tabel mewakili label aktual. Sedangkan kolom, mewakili hasil prediksi. Pada kasus ini, terdapat 34 data yang sebenarnya positif dan berhasil diprediksi dengan benar sebagai positif (TP = 34). Selain itu, tidak ada data positif yang salah diprediksi sebagai negatif (FN = 0),

yang menunjukkan model sangat baik sangat baik dalam mengenali data positif. Namun terdapat 7 data yang sebenarnya negative tetapi salah diprediksi sebagai positif ( $FP = 7$ ), menandakan masih ada kekeliruan model dalam membedakan data negatif. Sementara itu, 12 data yang sebenarnya negatif berhasil di prediksi dengan benar sebagai negative ( $TN = 12$ ). Secara keseluruhan matriks ini menggambarkan bahwa model memiliki kinerja yang sangat baik dalam mendeteksi data positif, namun perlu sedikit perbaikan dalam mengklasifikasi data negative. Kemudian, hasil dari prediksi di atas dapat digunakan untuk menghitung nilai akurasi, presisi, *recall*, dan *F1-Score* sebagai berikut:

1. Nilai akurasi

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + TN + FP + FN} \\
 &= \frac{34 + 12}{34 + 12 + 7 + 0} \\
 &= \frac{46}{53} \\
 &= 0.86
 \end{aligned}$$

2. Nilai Presisi

$$\begin{aligned}
 \text{Presisi} &= \frac{TP}{TP + FP} \\
 &= \frac{34}{34 + 7} \\
 &= \frac{34}{41} \\
 &= 0.82
 \end{aligned}$$

3. Nilai *Recall*

$$\text{recall} = \frac{TP}{TP + FN}$$

$$= \frac{34}{34 + 0}$$

$$= 1$$

#### 4. *F1-Score*

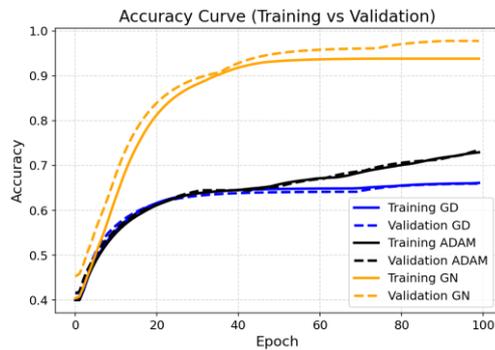
$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

$$= 2 \times \frac{0.82 \times 1}{0.82 + 1}$$

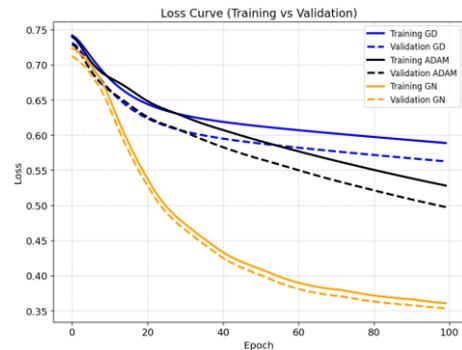
$$= 0.9067$$

Berdasarkan hasil pengujian terhadap dataset *Ionosphere*, model menunjukkan performa yang sangat baik dalam mendeteksi sinyal. Nilai akurasi sebesar 0.8679 mengidentifikasi bahwa secara keseluruhan, model mampu melakukan klasifikasi dengan Tingkat ketepatan yang tinggi. Nilai presisi sebesar 0.8293 menunjukkan bahwa Sebagian besar prediksi positif yang dihasilkan oleh model benar-benar merupakan data positif. Sehingga Tingkat kesalahan dalam prediksi positif relative rendah. Sementara itu, nilai *recall* sebesar 1 menandakan bahwa model memiliki sensitivitas yang sempurna, yakni seluruh data aktual positif berhasil mendeteksi dengan benar tanpa adanya kesalahan (*false negative*). Adapun nilai *F1-Score* sebesar 0.9067 memperlihatkan bahwa model mencapai keseimbangan yang sangat baik antara presisi recall. Oleh karena itu, dapat disimpulkan bahwa model yang dikembangkan memiliki kinerja yang andal dan efektif dalam tugas klasifikasi sinyal ppada dataset *ionosphere*.

#### 4.9 Kurva Perbandingan dengan Metode Optimasi Dasar



Gambar 4.5 Perbandingan kurva akurasi pada data Training dan Validasi Antara Metode Dasar GN, ADAM, GD



Gambar 4.4 Perbandingan Loss Data Training dan Validasi antara Metode GN, GD, dan ADAM

Berdasarkan hasil eksperimen terbukti bahwa metode Gauss-Newton lebih unggul dibandingkan dengan metode optimasi dasar seperti *gradient descent* dan ADAM, khususnya dalam konteks klasifikasi data Ionosphere. Dari gambar 4.2 GN mampu menghasilkan nilai loss yang lebih rendah dan stabil dibandingkan GD. Meskipun kecepatan penurunan loss GN sedikit lebih lambat dibandingkan ADAM pada tahap awal penelitian. GN menunjukkan konsistensi yang lebih baik menuju konvergensi, dengan variabilitas loss yang lebih kecil antara data pelatihan dan validasi. Hal ini mengidentifikasi bahwa GN memiliki kemampuan generalisasi yang lebih baik dan lebih tahan terhadap *overfitting*.

Berdasarkan gambar 4.3, GN mencapai akurasi validasi yang tinggi secara bertahap dan stabil, hamper menyamai bahkan dalam beberapa titik melampaui performa ADAM. Dibandingkan dengan GD, peningkatan performa GN jauh lebih signifikan, dimana GD mengalami kesulitan mencapai akurasi di atas 0.8 meskipun telah di latih selama 100 epoch.

Hal ini menunjukkan bahwa metode GN, dengan pendekatan berbasis pendekatan kuadrat terkecil terhadap fungsi kesalahan, mampu menangkap struktur

hubungan nonlinier dari pada metode GD yang hanya mengandalkan gradient biasa. Jika dibandingkan ADAM, GN lebih unggul dalam hal kestabilan pelatihan dan performa validasi yang lebih seimbang, tanpa mengalami fluktuasi akurasi yang besar. Oleh karena itu, dapat disimpulkan bahwa metode GN memberikan hasil optimasi yang lebih akurat dibandingkan metode GD dan ADAM.

#### 4.10 Perbandingan dengan Metode Dasar

Tabel 4.8 Perbandingan Performa dengan Menggunakan Metode GN, ADAM, dan GD pada Dataset Ionsphere

Metode & Autor	Akurasi	Precc	Rec	F1-Score	Time (ms)
GN	0.8679	0.8293	1.000	0.9067	570
ADAM	0.6981	0.6800	1.000	0.8095	431
GD	0.6792	0.6667	1.000	0.8	555

Berdasarkan hasil pengujian terhadap data Ionsphere menggunakan metode dasar ADAM dan GD, terbukti bahwa metode Gauss-Newton terbukti memberikan performa terbaik dibandingkan dengan metode ADAM maupun GD. Hal ini terlihat dari nilai matrix evaluasi model yang dihasilkan. Metode Gauss-Newton menghasilkan akurasi sebesar 86.79% yang menunjukkan model ini paling tepat dalam menghasilkan data secara keseluruhan. Selain itu, *recall* yang dihasilkan mencapai 100%, artinya semua data yang harusnya diklasifikasi sebagai positif berhasil dikenali seluruhnya tanpa ada kesalahan. Nilai *F1-Score* metode Gauss-Newton juga paling tinggi yaitu 0.9067, yang menandakan keseimbangan yang sangat baik antara presisi dan recall.

Jika dibandingkan dengan Gauss-Newton, metode ADAM menghasilkan akurasi 0.6981 dan *F1-Score* sebesar 0.68, yang cukup baik namun sedikit lebih rendah. Sementara itu, metode *gradient descent* menunjukkan performa yang paling rendah, dengan akurasi hanya 0.6792 dan *F1-Score* sebesar 0.8. Hal ini

menunjukkan bahwa metode *gradient descent* kurang efektif dalam menemukan parameter optimal pada data ini. Dari segi waktu, ADAM merupakan metode yang sedikit lebih unggul 431 ms, kemudian diusul oleh GD 555 ms, dan yang terakhir GN 570 ms. Secara keseluruhan, dapat disimpulkan bahwa pendekatan metode Gauss-Newton lebih unggul karena mampu mengoptimalkan parameter model dengan lebih cepat dan akurat dan menghasilkan performa klasifikasi lebih baik dibandingkan dua metode dasar lainnya dengan keunggulan pada *accuracy*, *Precision*, *recall*, *F1-score*, serta waktu.

#### 4.11 Perbandingan Hasil Penelitian dengan Penelitian Sebelumnya

Untuk menilai kinerja model dalam penelitian ini, dilakukan perbandingan dengan hasil-hasil penelitian terdahulu yang juga menggunakan dataset *Ionosphere*. Tabel berikut merangkum perbandingan akurasi, presisi, *recall*, *F1-Score*, serta beberapa metrik lainnya berdasarkan berdasarkan metode dan studi pada tabel 4.7

Tabel 4.9 Perbandingan dengan Penelitian Sebelumnya

Autor	Metode	<i>Acc</i>	<i>Precc</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Time (ms)</i>
Penelitian ini	GN	0.8679	0.8293	1.000	0.9067	570
(Aryani & Wijayanto, 2021)	SVM	0.7547	0.6176	1.000	-	-
	Naïve Bayes	0.7736	0.6765	0.6765	-	-
	<i>Random Forest</i>	0.9057	0.9412	0.9143	-	-
(Sigillito et al., 1989)	Single layer Network	0.92	-	-	-	-
	MLFN	0.98	-	-	-	-
(Saputra, et al., 2020)	SVM	0.836	-	-	0.90	-
	<i>Naïve Bayes</i>	0.835	-	-	0.757	-
	CART	0.86	-	-	0.965	-

Be(Aryani & Wijayanto, 2021), yaitu mencapai 1, yang menunjukkan keunggulan dalam mendeteksi semua ke berdasarkan tabel 4.10, model

dikembangkan dalam penelitian ini menggunakan metode GN menunjukkan performa yang dengan akurasi sebesar 0.8679, presisi sebesar 0.8293, *recall* sebesar 1, dan *F1-Score* sebesar 0.9067. Jika dibandingkan dengan hasil (Aryani & Wijayanto, 2021), model Gauss Newton dalam penelitian ini memiliki akurasi lebih tinggi dari pada SVM (0.7547) dan *Naïve Bayes* (0.7736), namun sedikit lebih rendah dibandingkan dengan *Random Forest* yang mencapai akurasi 0.9057. Selain itu, nilai *recall* model ini sama dengan SVM milik las positif.

Sementara itu, jika dibandingkan dengan (Sigillito et al., 1989) yang menggunakan metode *single layer network* dan MLFN diketahui bahwa akurasi pengujian cukup tinggi. Namun pada penelitian (Sigillito et al., 1989), tidak dilaporkan secara langsung presisi, *recall*, *F1-score*, sehingga lebih unggul Gauss Newton karna lebih lengkap evaluasi modelnya. Terakhir, hasil penelitian (Saputra et al., 2020) yang mengimplementasikan metode SVM, *Naïve Bayes*, dan CART juga menunjukkan bahwa CART memiliki akurasi terbaik sebesar 0.860 yang hamper setara dengan metode Gauss-Newton yang dikembangkan dalam penelitian ini.

Secara keseluruhan, dapat disimpulkan bahwa metode Gauss-Newton yang diusulkan dalam penelitian ini mampu menghasilkan kinerja klasifikasi yang sangat baik, kompetitif terhadap metode-metode populer seperti SVM, *Random Forest*, maupun jaringan saraf sederhana, dengan keunggulan utama terletak pada tingkat *recall* yang sempurna 1 dan keseimbangan presisi yang baik.

#### **4.12 Kajian Islam**

Perkembangan teknologi dan ilmu pengetahuan di era modern menghadirkan kemajuan luar biasa dalam aspek kehidupan. Namun, di balik kemajuan ini muncul

tantangan baru, terutama dalam hal pengolahan data yang akurat, etis, dan bertanggung jawab. Di dunia yang serba digital ini, data memegang peranan penting sebagai dasar dalam pengambilan keputusan di berbagai sektor, mulai dari kesehatan, ekonomi, hingga keamanan. klasifikasi biner yang menggunakan regresi logistik merupakan salah satu bentuk aplikasi pengolahan data untuk memisahkan atau mengelompokkan objek ke dalam dua kategori berbeda. Dalam konteks ini, keakuratan data menjadi faktor tertentu yang sangat krusial. Data yang tidak valid, dimanipulasi, atau tidak memiliki realitas dapat menyebabkan kesalahan dalam hasil klasifikasi dan pada akhirnya merugikan banyak pihak.

Dalam islam, nilai-nilai kejujuran, keadilan, dan akuntabilitas memiliki posisi yang sangat fundamental dan menjadi pilar utama dalam setiap aspek kehidupan, termasuk dalam aktivitas ilmiah seperti pengolahan data. Islam mengajarkan bahwa segala perbuatan manusia tidak hanya di nilai dari sisi duniawi, tetapi juga akan dipertanggung jawabkan di akhirat kelak. Oleh karena itu, proses analisis dan pengolahan data harus di landasi oleh nilai-nilai spiritual dan moral yang tinggi. Hal ini selaras dengan QS. Az-Zumar ayat 70.

أَوْفَيْتُ كُلَّ نَفْسٍ مَّا عَمِلَتْ وَهُوَ أَعْلَمُ بِمَا يَفْعَلُونَ ﴿٧٠﴾

(Kementerian Agama, 2022)

Artinya:” Dan tiap-tiap jiwa diberi balasan yang sempurna yang sesuai dengan apa yang telah dikerjakan, dan mereka tidak di zalimi”. (Q.S Az-Zumar [39]: 70))

Ayat ini mengandung makna yang sangat mendalam terkait dengan prinsip keadilan dan akuntabilitas dalam islam. Allah SWT menegaskan bahwa setiap amal manusia akan dibalas secara adil sesuai dengan apa yang telah manusia perbuat.

Jika seorang peneliti mengabaikan nilai kebenaran, seperti memanipulasi data, atau menyajikan hasil yang tidak sesuai dengan pesan moral dalam ayat ini.

Dalam QS. Az-Zumar: 70 Allah SWT menyampaikan bahwa “tidak ada yang dirugikan sedikitpun”, sebuah pengingat bahwa keadilan illahi bersifat *absolut*. Hal ini mengandung pelajaran bahwa dalam dunia akademik dan penelitian, hasil yang sah dan bermakna hanya bisa diperoleh dari jalan yang benar. Apabila ada pihak yang dirugikan akibat analisis data yang tidak benar, misalnya pasien yang salah dalam diagnosis karena klasifikasi yang keliru, atau kebijakan publik yang salah arah karena data tidak akurat, maka secara moral peneliti turut menanggung dampak kesalahan tersebut.

Kalimat penutup dalam ayat ini, “sungguh Allah SWT sangat cepat perhitungan-Nya”, juga memiliki implikasi mendalam. Di dunia yang serba cepat dan instan, terkadang seseorang merasa bahwa kesalahan kecil bisa “disembunyikan” karena tidak ada yang melihat. Namun, ayat ini menekankan bahwa Allah SWT tidak memerlukan waktu lama untuk menghitung dan mencatat semua perbuatan manusia sekecil apapun itu.

Dengan demikian, QS Az-Zumar: 70 memberikan fondasi moral dan spiritual yang kokoh bagi setiap peneliti dan ilmuan muslim. Ayat ini menanamkan kesadaran bahwa pekerjaan ilmiah bagian dari amanah yang harus dijalankan dengan penuh integritas. Kejujuran dalam menampilkan data, akurasi dalam perhitungan, serta transparansi dalam pelaporan bukan hanya memenuhi standar akademik, tetapi juga merupakan bentuk ibadah dan ketundukan terhadap prinsip keadilan Allah SWT.

Kegiatan analisis data bukanlah aktivitas yang bersifat netral, melainkan syarat dan tanggung jawab moral. Ketika seorang peneliti memilih untuk merubah data agar hasil lebih baik, ia sebenarnya sedang mengkhianati amanah serta menyalahi nilai-nilai islam. Proses pengolahan data etis dapat menjadi cerminan dari prinsip hisab yang akan di alami oleh setiap manusia di akhirat. Seperti halnya ayat sbelumnya bahwa disebutkan bahwa “tidak ada satupun yang luput dalam perhitungan”.

Selain itu, ketakwaan seseorang peneliti kepada Allah SWT menjadi Kompas moral dalam menjaga integritas ilmiah. Dalam QS. At-Taubah: 119, Allah SWT berfirman

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَكُونُوا مَعَ الصَّادِقِينَ ﴿١١٩﴾

(Kementerian Agama, 2022)

Artinya: “Wahai orang-orang yang beriman, bertakwalah kepada Allah SWT dan jadilah Bersama orang-orang yang jujur.” (Q.S At-Taubah [9]: 119)

Ayat ini menunjukkan bahwa keberhasilan dan keberkahan akan diberikan kepada orang-orang yang bertakwa, yakni mereka yang senantiasa menjaga kejujuran dan bertindak adil dalam segala urusan, termasuk dalam penelitian ilmiah. Takwa bukan berarti sekedar rasa takut, melainkan sikap sabar dan bertanggung jawab atas segala hal yang dilakukan. Dalam konteks pengolahan data, ketakwaan ini menjelma menjadi konteks kehati-hatian, ketelitian dalam memproses dan melaporkan data tanpa adanya menipu dan menyembunyikan kebenaran.

Penelitian ini sendiri bertujuan untuk meningkatkan akurasi dalam klasifikasi biner dengan menggunakan pendekatan optimasi melalui metode Gauss-Newton dan conjugate gradient pada regresi logistik. Keduanya merupakan teknik numerik

yang bertujuan untuk memaksimalkan efisiensi dan akurasi dalam menentukan parameter model. Dengan demikian, penelitian ini tidak hanya memperkuat performa teknis dari metode biner yang digunakan, tetapi juga menjadi bentuk implementasi nyata dari nilai-nilai spiritual dalam proses ilmiah.

Metode Gauss-Newton dan conjugate gradient dalam regresi logistik adalah alat bantu yang kuat dalam menghasilkan model klasifikasi yang presisi. Namun, kekuatan teknis ini tidak akan berarti apa-apa tanpa kejujuran dalam integritas di setiap langkahnya. Menjaga kejujuran dan pengumpulan data, transparansi dalam pengolahan, serta tanggung jawab dan pelaporan adalah bentuk ibadah tersendiri dari seorang peneliti muslim. Maka dari itu, proses penelitian bukan hanya sekedar rangkaian teknis, tetapi juga merupakan cerminan dari keimanan dan ketakwaan kepada Allah SWT yang harus senantiasa dijaga dalam setiap tahapannya.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan rumusan masalah dan pembahasan pada bab sebelumnya, maka diperoleh Kesimpulan sebagai berikut:

1. Implementasi metode Gauss-Newton dengan conjugate gradient pada pelatihan regresi logistik untuk klasifikasi biner menunjukkan performa yang unggul dibandingkan metode Gradient Descent dan Adam. Pada epoch ke-100, metode ini berhasil mencapai validation accuracy sebesar 97,78%, lebih tinggi dibandingkan Adam sebesar 73,42% (selisih +24,36%) dan Gradient Descent sebesar 65,95% (selisih +31,83%). Nilai training loss yang dihasilkan juga lebih rendah, yaitu 0,3603, sementara Adam berada di 0,5280 (selisih -0,1677) dan Gradient Descent di 0,5886 (selisih -0,2283 terhadap Gauss-Newton). Proses penurunan loss pun lebih cepat, di mana pada epoch ke-60, Gauss-Newton sudah mencapai training loss 0,3912, sedangkan Adam masih di 0,5514 (selisih -0,1602) dan Gradient Descent di 0,5825 (selisih -0,1913). Selisih akurasi yang cukup signifikan sebesar +24,36% terhadap Adam dan +31,83% terhadap Gradient Descent, serta selisih training loss yang konsisten lebih rendah, membuktikan bahwa conjugate gradient dalam metode Gauss-Newton mampu mempercepat proses konvergensi dan menghasilkan model yang lebih stabil serta akurat dalam pelatihan regresi logistik biner.
2. Penerapan conjugate gradient dalam metode Gauss-Newton terbukti mampu meningkatkan akurasi dan efisiensi prediksi dalam klasifikasi biner. Metode ini berhasil mencapai akurasi sebesar 86,79%, lebih tinggi dibandingkan Gradient Descent dengan selisih +18,87% dan Adam sebesar +16,98%. Jika dibandingkan dengan penelitian terdahulu, metode ini juga unggul atas

beberapa metode lainnya, yaitu lebih tinggi dibandingkan SVM milik Aryani sebesar +11,32%, Naïve Bayes milik Aryani sebesar +9,43%, dan SVM milik Saputra sebesar +3,19%. Meskipun Random Forest milik Aryani memiliki akurasi yang lebih tinggi dengan selisih -3,78% dibandingkan metode ini, namun metode Gauss-Newton unggul dalam efisiensi waktu dan stabilitas hasil. Dibandingkan metode Single Layer Network dan MLFN milik Sigillito yang masing-masing lebih tinggi sebesar -5,21% dan -11,21%, metode ini tetap menawarkan kelebihan pada kecepatan konvergensi dan nilai recall sempurna sebesar 1,000 yang tidak dilaporkan pada metode-metode tersebut. Selain itu, conjugate gradient mempercepat proses penurunan training loss hingga di bawah 0,40 (tepatnya 0,3603) dalam waktu 570 ms, sedangkan metode lain membutuhkan waktu lebih lama atau bahkan tidak mencapai nilai loss serupa dalam jumlah epoch yang sama. Seluruh selisih performa ini menunjukkan bahwa conjugate gradient pada metode Gauss-Newton mampu memberikan peningkatan signifikan dari segi akurasi, kecepatan konvergensi, recall sempurna, serta efisiensi prediksi dibandingkan berbagai metode yang telah dilaporkan dalam penelitian terdahulu..

## 5.2 Saran

Berdasarkan hasil penelitian yang diperoleh, terdapat beberapa saran yang dapat diberikan untuk mengembangkan penelitian selanjutnya, maka penulis memberikan saran berikut:

1. Perlu dilakukan pengujian metode Gauss-Newton dengan conjugate gradient pada dataset lain dengan karakteristik yang berbeda, baik dari segi jumlah

fitur, distribusi data, maupun ukuran dataset, untuk melihat konsistensi performa dan keandalan metode ini dalam berbagai kasus klasifikasi biner.

2. Penelitian selanjutnya dapat membandingkan Gauss-Newton dengan algoritma optimasi lain seperti *Lavenberg Marquard* atau BFGS dalam pelatihan model regresi logistik, sehingga diperoleh Gambaran yang lebih luas mengenai keunggulan relatif dari masing-masing metode, baik dari *accuracy*, *precision*, *recall*, *F1-Score* maupun waktu.
3. Penting pula untuk melakukan pengujian terhadap performa metode ini pada data yang tidak seimbang untuk mengetahui sejauh mana kemampuan metode Gauss-Newton dengan conjugate gradient dalam mempertahankan *recall* yang sempurna, serta keseimbangan matrixs evaluasi lainnya.

## DAFTAR PUSTAKA

- Afifah, devi nur. (2020). *perbandingan metode regresi logistik biner pada kesejahteraan rumah tangga di kabupaten Mojokerto*.
- Aryani, Y., & Wijayanto, A. W. (2021). Klasifikasi Pengembalian Radar dari Ionosfer Menggunakan SVM, Naïve Bayes dan Random Forest. *Komputika : Jurnal Sistem Komputer*, 10(2), 111–117.  
<https://doi.org/10.34010/komputika.v10i2.4347>
- Barnett, V., Neter, J., & Wasserman, W. (2005). Applied Linear Statistical Models. In *Journal of the Royal Statistical Society. Series A (General)* (Vol. 138, Issue 2). <https://doi.org/10.2307/2984653>
- Bishop, C. M. (2006). A Review of Pattern Recognition and Machine Learning. In *Journal of Machine and Computing* (Vol. 4, Issue 1).  
<https://doi.org/10.53759/7669/jmc202404020>
- Bjorck, A. (1996). *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics (SIAM).
- Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- Dembo, R. S., Eisenstat, S. C., & Steihaug, T. (1982). Inexact Newton Methods. *SIAM Journal on Numerical Analysis*, 19(2), 400–408.  
<https://doi.org/10.1137/0719025>
- Fletcher, R. (1987). *Practical Methods of Optimization*. (2nd (ed.)). Wiley.
- Hastie et al. (2009). 18. High-Dimensional Problems:  $p \gg N$ . In *The elements of statistical learning: data mining, inference and prediction* (Vol. 27, Issue 2).
- Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 409. <https://doi.org/10.6028/jres.049.044>
- Hosmer, et al. (2013). *Applied Logistic Regression* ((3rd.ed) (ed.)). Wiley.
- Hosmer and Lemeshow. (1991). *Applied Logistic Regression-Wiley-Series-in-Probab.Pdf*.
- Jamhuri, M., Mukhlash, I., & Irawan, M. I. (2022). Performance Improvement of Logistic Regression for Binary Classification by Gauss-Newton Method. *ACM International Conference Proceeding Series*, 12–16.  
<https://doi.org/10.1145/3545839.3545842>
- Kementerian Agama. (2022). *Qur'an Kemenag*
- Kouh, et all., 2008. (2008). A Canonical Neural Circuit for Cortical Nonlinear Operations. *Neural Computation*, 20(6), 1427–1451.

<https://doi.org/10.1162/neco.2008.03-07-532>

- Lan, G. (2004). *Convex Optimization Theory*. [https://doi.org/10.1007/978-3-030-39568-1\\_2](https://doi.org/10.1007/978-3-030-39568-1_2)
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Neuhaus, J., & McCulloch, C. (2010). Generalized linear models. In *Wiley Interdisciplinary Reviews: Computational Statistics* (Vol. 3, Issue 5). <https://doi.org/10.1002/wics.175>
- Nocedal, J., & Wright, S. J. (2006). *Numerical Optimization*. Springer.
- Nocedal, J., & Wright, S. (2006). Numerical Optimization Second Edition. In *Ear and Hearing* (Vol. 17, Issue 2).
- Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1–2), 1–135. <https://doi.org/10.1561/1500000011>
- Prasad, N. R., Almanza-Garcia, S., & Lu, T. T. (2009). Anomaly detection. *Computers, Materials and Continua*, 14(1), 1–22. <https://doi.org/10.1145/1541880.1541882>
- Rendle, S. (2012). *Factorization Machines with LIBFM*. 1232–1237. <https://doi.org/10.1109/ICDM.2012.181>
- Saputra, D., Dharmawan, W. S., Wahyudi, M., Irmayani, W., Sidauruk, J., & Martias. (2020). Performance Comparison and Optimized Algorithm Classification. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012087>
- Sauer, T. (2017). Numerical Analysis. In *Journal of Physics A: Mathematical and Theoretical* (Vol. 44, Issue 8). <https://math.gmu.edu/~tsauer/>
- Scott, A. J., Hosmer, D. W., & Lemeshow, S. (1991). Applied Logistic Regression. In *Biometrics* (2nd ed., Vol. 47, Issue 4). Wiley. <https://doi.org/10.2307/2532419>
- Shewchuk, J. R. (1994). *An introduction to the conjugate gradient method without the agonizing pain*. Laporan Shewchuk di arXiv
- Sigillito, V. G., Wing, S. P., Hutton, L. V., & Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 10(3), 262–266.
- Sutrisno, E. (2020). *Aplikasi quasicomplete separation dengan metode maximum likelihood pada regresi logistik*.
- Umaroh, A. (2020). *Perbandingan metode regresi logistik biner dan clasification and regression tree pada klasifikasi status kesejahteraan rumah tangga di*

*kota batu.*

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), 301–320. <https://doi.org/10.1111/j.1467-9868.2005.00503.x>

## LAMPIRAN

### Lampiran 1 Data Ionosphere



#### Contoh Data

No	X								Y
	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	...	$x_{34}$	$y$
1	1	0	0.9953	-0.0588	0.8524	0.0230	...	-0.453	<i>g</i>
2	1	0	1	-0.1882	0.9303	-0.3615	...	-0.0244	<i>b</i>
3	1	0	1	-0.0336	1	0.0048	...	-0.3823	<i>g</i>
4	1	0	1	-0.4516	1	1	...	1	<i>b</i>
5	1	0	1	-0.0240	0.9414	0.0653	...	-0.6569	<i>g</i>
6	1	0	0.0233	-0.0059	-0.0992	-0.1194	...	0.1201	<i>b</i>
7	1	0	0.9758	-0.060	0.9460	-0.0208	...	-0.8097	<i>g</i>
8	1	0	0	0	0	0	...	0	<i>b</i>
9	1	0	0.9635	-0.0719	1	-0.1433	...	-0.7342	<i>g</i>
10	1	0	-0.018	-0.0845	0	0	...	0.3805	<i>b</i>
11	1	0	1	0.0665	1	-0.1838	...	-0.8915	<i>g</i>
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
351	1	0	0.8471	0.7363	0.7363	-0.061	...	-0.0615	<i>g</i>

## Lampiran 2 Hasil Normalisasi Seluruh Data

	0	1	2	3	4	5	6	7	\
0	1.0	1.0	0.0	0.997695	0.470555	0.926215	0.511530	0.916990	
1	1.0	1.0	0.0	1.000000	0.405855	0.965175	0.319220	0.445660	
2	1.0	1.0	0.0	1.000000	0.483175	1.000000	0.502425	1.000000	
3	1.0	1.0	0.0	1.000000	0.274195	1.000000	1.000000	0.856080	
4	1.0	1.0	0.0	1.000000	0.487995	0.970700	0.532655	0.960530	
..	...	...	...	...	...	...	...	...	
346	1.0	1.0	0.0	0.917540	0.541490	0.868695	0.426470	0.921745	
347	1.0	1.0	0.0	0.975565	0.502095	0.975915	0.486385	0.967190	
348	1.0	1.0	0.0	0.973505	0.499830	0.966035	0.483865	0.975885	
349	1.0	1.0	0.0	0.953040	0.491715	0.990610	0.490055	0.978455	
350	1.0	1.0	0.0	0.923550	0.567665	0.868190	0.469245	0.939365	
		8	9	...	25	26	27	28	\
0	0.311460	1.000000	...	0.784055	0.244145	0.705390	0.269160		
1	0.032015	1.000000	...	0.398340	0.367155	0.397660	0.407995		
2	0.439690	0.944825	...	0.787640	0.298900	0.794920	0.389275		
3	0.000000	0.500000	...	1.000000	0.953475	0.758065	1.000000		
4	0.383725	0.885760	...	0.516430	0.174210	0.566450	0.233970		
..	...	...	...	...	...	...	...		
346	0.472165	0.952205	...	0.976890	0.478990	0.917395	0.500615		
347	0.490400	0.972950	...	0.972600	0.506805	0.967610	0.524625		
348	0.482845	0.977920	...	0.969940	0.515965	0.962445	0.512710		
349	0.481770	0.928730	...	0.955250	0.489505	0.945735	0.461200		
350	0.541300	0.944640	...	0.932335	0.424430	0.905735	0.475890		
		29	30	31	32	33	34		
0	0.606330	0.329550	0.711335	0.227565	0.593205	0.273500			
1	0.404800	0.442035	0.416870	0.468560	0.431310	0.487765			
2	0.715500	0.413175	0.802180	0.379100	0.780225	0.308810			
3	1.000000	0.399505	0.628410	1.000000	0.338090	1.000000			
4	0.512155	0.189015	0.471465	0.202135	0.476960	0.171515			
..	...	...	...	...	...	...			
346	1.000000	0.564075	0.933300	0.446430	0.952730	0.478465			
347	0.965795	0.540840	0.970330	0.499825	0.957415	0.523560			
348	0.960600	0.511210	0.962295	0.502210	0.963485	0.497115			
349	0.914915	0.413810	0.980110	0.481215	0.937015	0.418785			
350	0.891035	0.496485	0.878735	0.466610	0.928820	0.469245			

[351 rows x 35 columns]

## Lampiran 3 Script Python Metode Gauss Newton

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.preprocessing import MinMaxScaler

np.random.seed(42)

# Load dataset Ionosphere
```

```

url = 'https://archive.ics.uci.edu/ml/machine-learning-
databases/ionosphere/ionosphere.data'
column_names = [f'Feature_{i}' for i in range(34)] + ['Label']
data = pd.read_csv(url, header=None, names=column_names)

data['Label'] = data['Label'].map({'g': 1, 'b': 0})
X = data.drop('Label', axis=1).values
y = data['Label'].values

# Normalisasi pakai MinMaxScaler
scaler = MinMaxScaler()
X = scaler.fit_transform(X)
X = np.c_[np.ones(data.shape[0]), X] # tambah bias term

# Data sebelum normalisasi (buat DataFrame dari data asli)
df_features = data.copy()
print("\nData sebelum normalisasi (5 data pertama):")
print(df_features.head())

# Split dataset
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3,
random_state=42, stratify=y)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42, stratify=y_temp)

print("\nHasil normalisasi (semua data):")
print(pd.DataFrame(X))

# Fungsi aktivasi sigmoid
def sigmoid(z):
    return 1 / (1 + np.exp(-z))

# Fungsi loss Binary Cross-Entropy
def loss(y, h):
    return -np.mean(y * np.log(h + 1e-8) + (1 - y) * np.log(1 - h + 1e-8))
# Fungsi gauss_newton untuk hitung matriks A dan vektor b serta output h
def gauss_newton(X, y, w):
    z = X @ w
    h = sigmoid(z)
    J = (y - h)[:, np.newaxis] * X
    A = J.T @ J
    b = J.T @ (-y * np.log(h + 1e-8) - (1 - y) * np.log(1 - h + 1e-8))
    return A, b, h

# Smoothing curve
def smooth_curve(data, weight=0.9, increasing=True):
    smoothed = [data[0]]

```

```

for point in data[1:]:
    if increasing:
        smoothed.append(max(smoothed[-1] * weight + (1 - weight) * point,
smoothed[-1]))
    else:
        smoothed.append(min(smoothed[-1] * weight + (1 - weight) * point,
smoothed[-1]))
    return smoothed

# Fungsi conjugate_gradient untuk training dengan metode conjugate gradient +
gauss_newton
def conjugate_gradient(X, y, X_val, y_val, max_epochs=200, tol=1e-6,
learning_rate=0.05):
    n_samples, n_features = X.shape
    w = np.random.randn(n_features) * np.sqrt(1 / n_features)
    losses, val_losses, accuracies, val_accuracies = [], [], [], []

    for epoch in range(max_epochs):
        A, b, h = gauss_newton(X, y, w)
        r = b - A @ w
        d = r.copy()

        for _ in range(n_features):
            Ad = A @ d
            alpha = (r.T @ r) / (d.T @ Ad + 1e-8)
            w_new = w + learning_rate * alpha * d
            r_new = r - learning_rate * alpha * Ad

            if np.linalg.norm(r_new) < tol:
                w = w_new
                break

            beta = (r_new.T @ r_new) / (r.T @ r + 1e-8)
            d = r_new + beta * d
            w, r = w_new, r_new

# Hitung loss dan akurasi train dan val
loss_train = loss(y, sigmoid(X @ w))
acc_train = np.mean((sigmoid(X @ w) >= 0.5).astype(int) == y)
loss_val = loss(y_val, sigmoid(X_val @ w))
acc_val = np.mean((sigmoid(X_val @ w) >= 0.5).astype(int) == y_val)

losses.append(loss_train)
val_losses.append(loss_val)
accuracies.append(acc_train)
val_accuracies.append(acc_val)

```

```

# Early stopping sederhana
if epoch > 10 and abs(losses[-1] - losses[-2]) < tol and val_accuracies[-1] <
val_accuracies[-2]:
    print(f"Early stopping at epoch {epoch+1}")
    break

return w, smooth_curve(losses, increasing=False), smooth_curve(val_losses,
increasing=False), \
    smooth_curve(accuracies, increasing=True),
smooth_curve(val_accuracies, increasing=True)

# Training model
w, losses_GN, val_losses_GN, accuracies_GN, val_accuracies_GN =
conjugate_gradient(
    X_train, y_train, X_val, y_val, max_epochs=100
)

# Prediksi
y_pred_prob = sigmoid(X_test @ w)
y_pred = (y_pred_prob >= 0.5).astype(int)

# Confusion Matrix & Evaluasi
cm = confusion_matrix(y_test, y_pred)
TP, TN, FP, FN = cm[1, 1], cm[0, 0], cm[0, 1], cm[1, 0]
accuracy = (TP + TN) / (TP + TN + FP + FN)
precision = TP / (TP + FP)
recall = TP / (TP + FN)
f1 = 2 * (precision * recall) / (precision + recall)

print(f'Confusion Matrix:\n{cm}')
print(f'TP: {TP}, TN: {TN}, FP: {FP}, FN: {FN}')
print(f'Accuracy: {accuracy:.4f}')
print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')

# Plot Confusion Matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.show()
%%time

# Plot Loss
plt.figure(figsize=(8,6))
plt.plot(losses_GN, label='Training Loss', linestyle='--', linewidth=2,
color='green')
plt.plot(val_losses_GN, label='Validation Loss', linestyle='--', linewidth=2,
color='purple')

```

```

plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Loss Curve (Training vs Validation)')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()

# Plot Accuracy
plt.figure(figsize=(8,6))
plt.plot(accuracies_GN, label='Training Accuracy', linewidth=3, color='green')
plt.plot(val_accuracies_GN, label='Validation Accuracy', linewidth=3,
color='purple')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Accuracy Curve (Training vs Validation)')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()

# Menampilkan hasil dalam bentuk tabel
results = []
for i in range(len(losses_GN)):
    results.append([i+1, losses_GN[i], val_losses_GN[i], accuracies_GN[i]*100,
val_accuracies_GN[i]*100])

df_results = pd.DataFrame(results, columns=['Epoch', 'Training Loss GN',
'Validation Loss GN', 'Training Accuracy GN', 'Validation Accuracy GN'])
print(df_results.to_string(index=False))

# Plot distribusi kelas
count_bad = np.sum(y == 0)
count_good = np.sum(y == 1)
counts = [count_good, count_bad]
labels = ['Good (1)', 'Bad (0)']
colors = ['blue', 'red']

fig, axes = plt.subplots(1, 2, figsize=(12, 5))

axes[0].bar(labels, counts, color=colors)
axes[0].set_xlabel('Class')
axes[0].set_ylabel('Count')
axes[0].set_title('Distribution of Classes')
axes[0].grid(axis='y', linestyle='--', alpha=0.6)

axes[1].pie(counts, labels=labels, autopct='%1.1f%%', colors=colors,
startangle=140,
wedgeprops={'edgecolor': 'black'})
axes[1].set_title('Class Distribution in Percentage')

```

```
plt.tight_layout()  
plt.show()
```

## RIWAYAT HIDUP



Silvi Puspita Sari, lahir di Tuban pada 25 Maret 2002. Penulis merupakan anak Tunggal dari Bapak Masrukin dan Ibu Lasmi. Pendidikan formal penulis dimulai di TK Dharma Wanita Pugoh, yang diselesaikan pada tahun 2009. Selanjutnya, penulis melanjutkan Pendidikan di SDN Pugoh 431 dan lulus pada tahun 2015. Pendidikan Tingkat menengah pertama ditempuh di SMPN 2 Bancar dan lulus pada tahun 2018. Penulis kemudian menyelesaikan Pendidikan menengah atas di MAN 1 Tuban pada tahun 2021. Pada tahun yang sama, penulis melanjutkan Pendidikan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang, dengan memilih Program Studi Matematika di Fakultas Sains dan Teknologi. Selama menempuh pendidikan di universitas penulis pernah menjadi anggota PSMGGB selama 2 periode. Penulis juga mengikuti beberapa kegiatan kepanitiaan yang diselenggarakan oleh DEMA Fakultas Saintek.



**KEMENTERIAN AGAMA RI**  
**UNIVERSITAS ISLAM NEGERI**  
**MAULANA MALIK IBRAHIM MALANG**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

### BUKTI KONSULTASI SKRIPSI

Nama : Silvi Puspita Sari  
NIM : 210601110060  
Fakultas / Jurusan : Sains dan Teknologi / Matematika  
Judul Skripsi : Peningkatan Performa Metode Gauss Newton dengan  
Conjugate Gradient pada Masalah Klasifikasi Biner  
Menggunakan Regresi Logistik  
Pembimbing I : Dr. Mohammad Jamhuri, M.Si.  
Pembimbing II : Ema Herawati, M.Pd.

No	Tanggal	Hal	Tanda Tangan
1.	25 September 2024	Konsultasi Topik dan Data	1.
2.	14 Oktober 2024	Konsultasi Bab I, II, dan III	2.
3.	21 Oktober 2024	Konsultasi Bab I, II, dan III	3.
4.	4 Desember 2024	Konsultasi Bab I, II, dan III	4.
5.	10 Desember 2024	Konsultasi Bab I, II, dan III	5.
6.	11 Desember 2024	ACC Bab I, II, dan III	6.
7.	12 Desember 2024	Konsultasi Kajian Agama Bab I dan II	7.
8.	4 Februari 2025	ACC Kajian Agama Bab I dan II	8.
9.	20 Februari 2025	ACC Seminar Proposal	9.
10.	15 April 2025	Konsultasi Revisi Seminar Proposal	10.
11.	18 April 2025	Konsultasi Bab IV dan V	11.
12.	25 April 2025	Konsultasi Bab IV dan V	12.
13.	7 Mei 2025	Konsultasi Bab IV dan V	13.
14.	12 Mei 2025	Konsultasi Bab IV dan V	14.



**KEMENTERIAN AGAMA RI  
UNIVERSITAS ISLAM NEGERI  
MAULANA MALIK IBRAHIM MALANG  
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

15.	13 Mei 2025	ACC Bab IV dan V	15. <i>[Signature]</i>
16.	13 Mei 2025	Konsultasi Kajian Agama Bab IV	16. <i>[Signature]</i>
17.	15 Mei 2025	ACC Kajian Agama Bab IV	17. <i>[Signature]</i>
18.	20 Mei 2025	ACC Seminar Hasil	18. <i>[Signature]</i>
19.	21 Mei 2025	ACC Seminar Hasil lanjutan	19. <i>[Signature]</i>
20.	10 Juni 2025	Konsultasi Revisi Seminar Hasil	20. <i>[Signature]</i>
21.	18 Juni 2025	ACC Sidang Skripsi	21. <i>[Signature]</i>
22.	18 Juni 2025	ACC Keseluruhan	22. <i>[Signature]</i>

Malang, 18 Juni 2025

Mengetahui,

Ketua Program Studi Matematika

*[Signature]*  
Dr. Elly Susanti, M.Sc.  
NIP. 19741129 200012 2 005